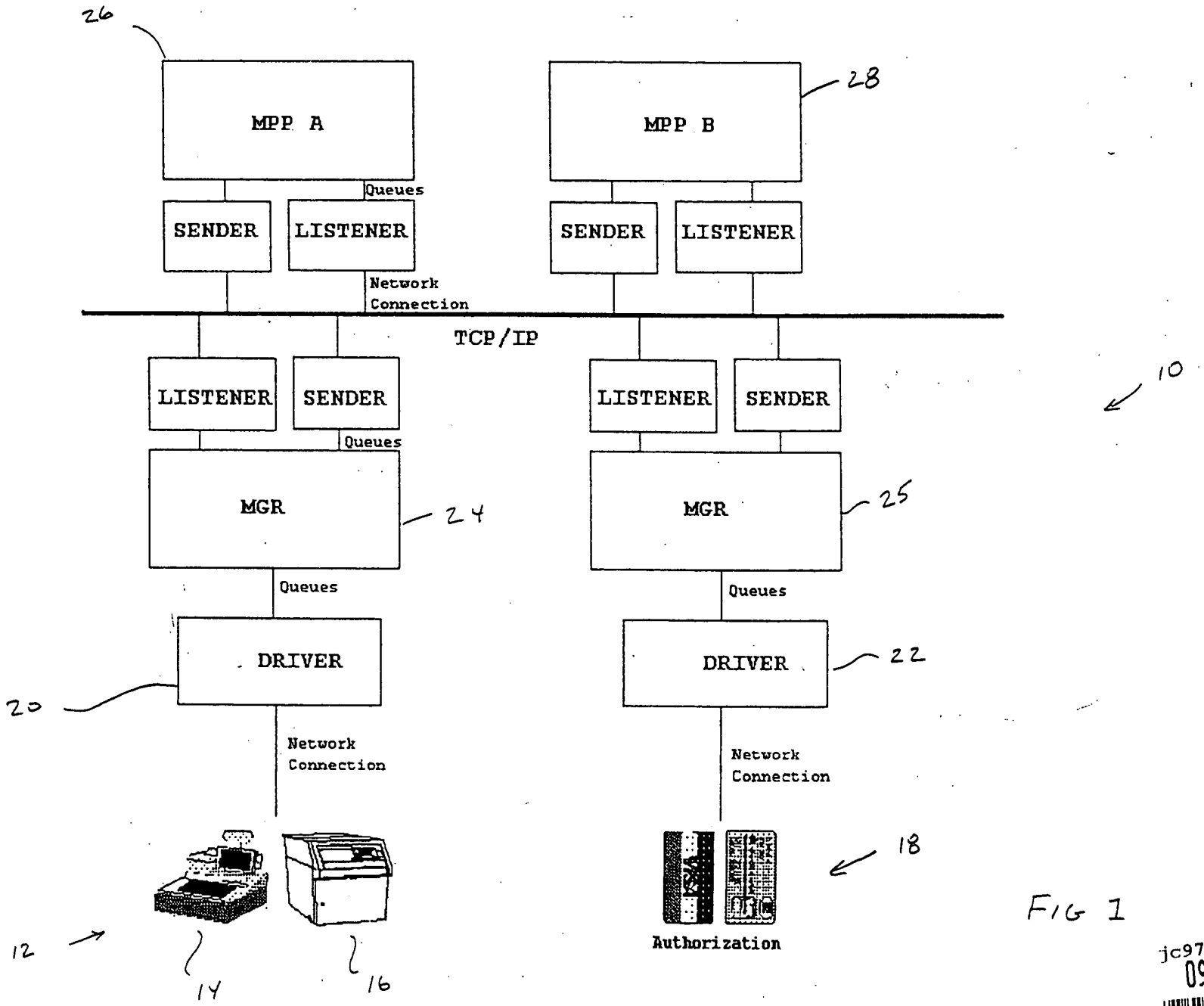
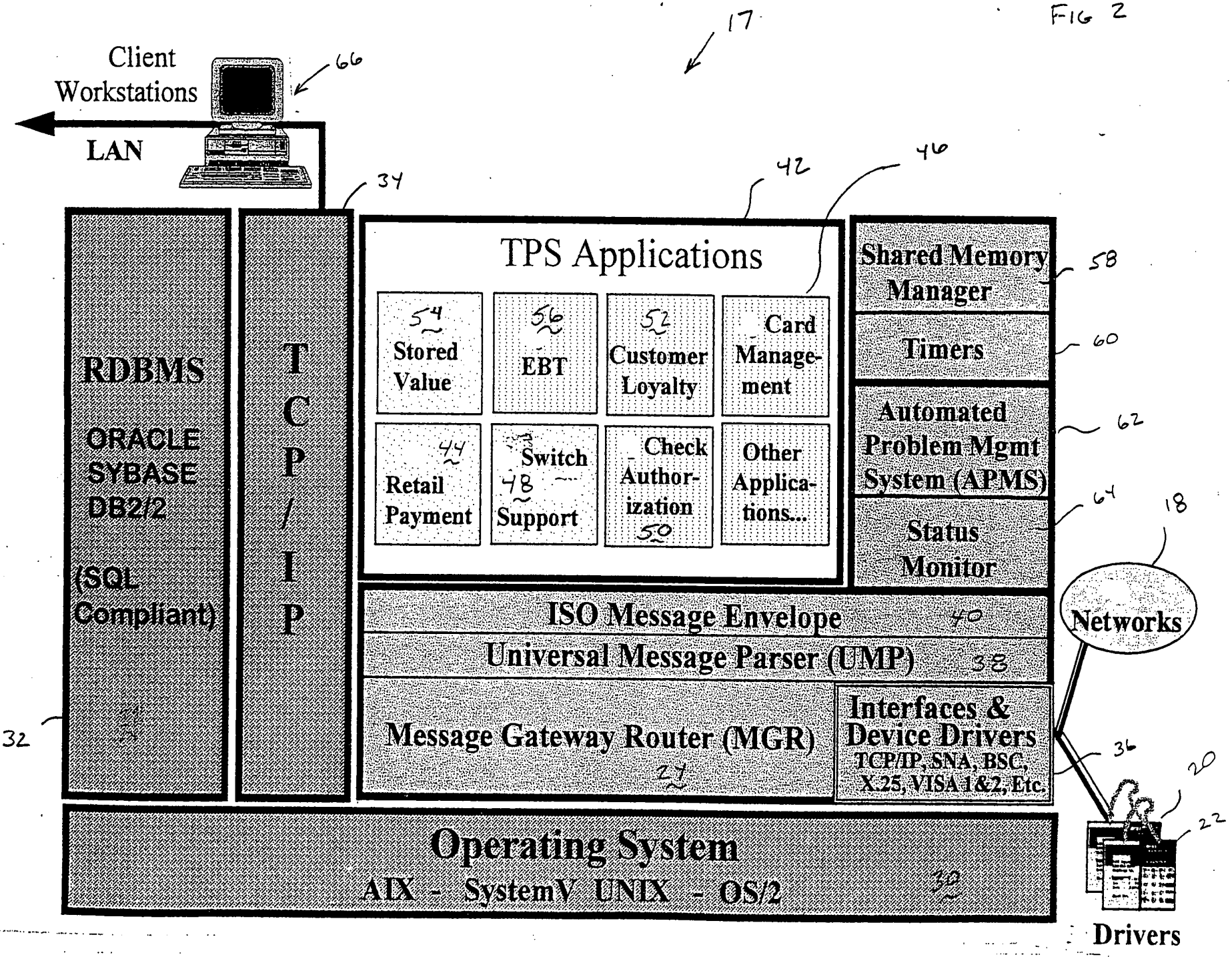


FIG 1

FIG 2


17
Client Workstations
66
LAN
34
42
46
RDBMS
ORACLE
SYBASE
DB2/2
(SQL Compliant)
32
TCP/IP
TPS Applications
54 Stored Value
56 EBT
52 Customer Loyalty
Card Management
44 Retail Payment
Switch
48 Support
Check Authorization
50
Other Applications...
Shared Memory Manager
58
Timers
60
Automated Problem Mgmt System (APMS)
62
Status Monitor
64
18
Networks
ISO Message Envelope
40
Universal Message Parser (UMP)
38
Message Gateway Router (MGR)
24
Interfaces & Device Drivers
TCP/IP, SNA, BSC, X.25, VISA1&2, Etc.
36
20
22
Drivers
Operating System
AIX - SystemV UNIX - OS/2
30

Standard Message Envelope (SME) Format.

1	Header Sid	Header Layout Version	1
2	Source Node Sid	The message originating node system Id.	6
3	Message Receive System Time	The system time in YYYYMMDDHHMISSmmm format.	17
4	Internal Message Sid	Unique system Id of the received message.	4
5	Service Sid	The Message Processing Program (MPP) service system Id, which can process received message.	4
6	Target Node Sid	The message receiving node system Id	6
7	Data Format Indicator (SOURCE)	Message data format type 0 - External Data SOURCE 1 - Internal Data SOURCE	1
8	Message Direction	The direction of message routing.	1
9	Processing Time	Elapsed message processing time in milliseconds.	5
10	Processing Node Sid	The last processing node system Id	6
11	Target Line Node Sid	Line driver node system id. Assigned when terminal is attached to line group.	6
12	Message Text	The message text in ISO8583 format	Variable

FIG 3

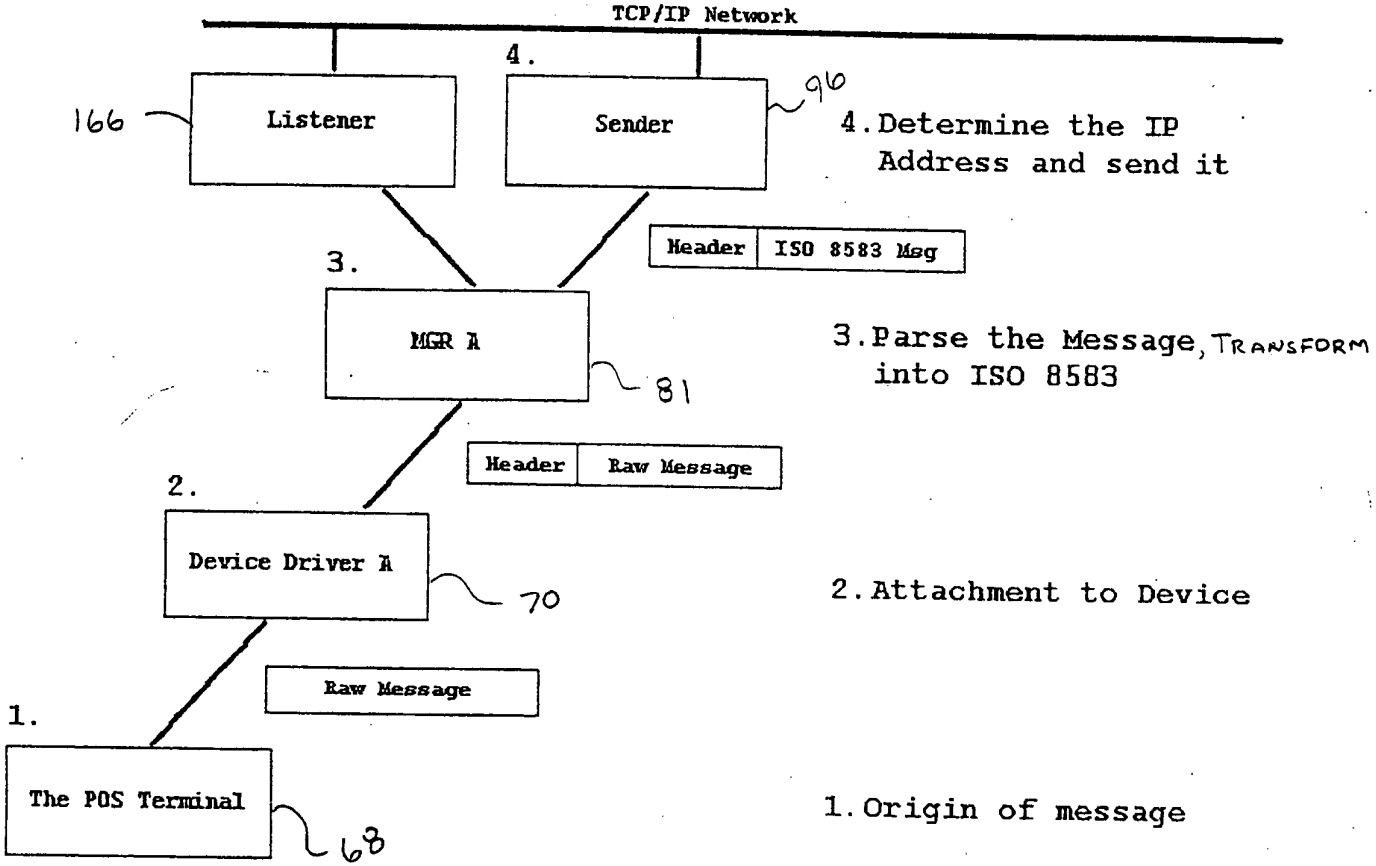
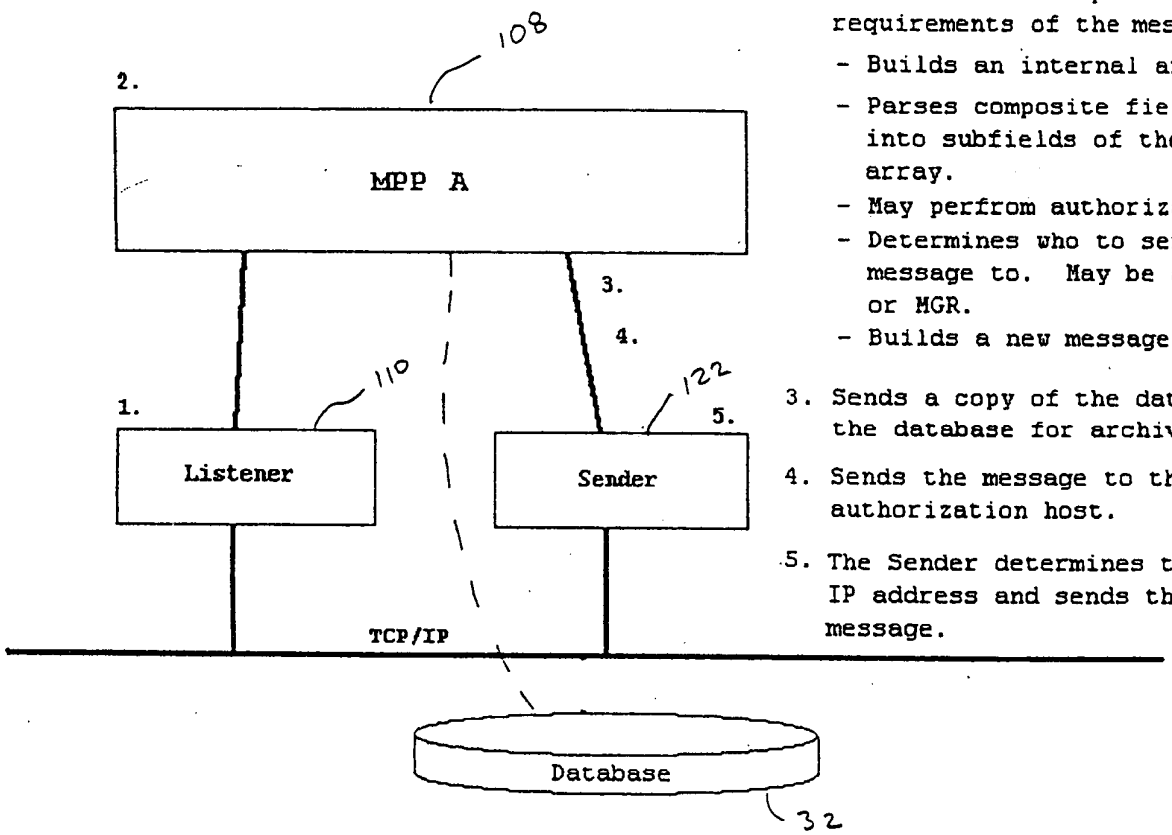

TCP/IP Network
4.
166
Listener
Sender
90
4. Determine the IP Address and send it
Header
ISO 8583 Msg
3.
MGR A
81
3. Parse the Message, TRANSFORM into ISO 8583
Header
Raw Message
2.
Device Driver A
70
2. Attachment to Device
Raw Message
1.
The POS Terminal
68
1. Origin of message

FIG 4



1. The Listener buffers the data, then places the data onto the input queue of the MPP.

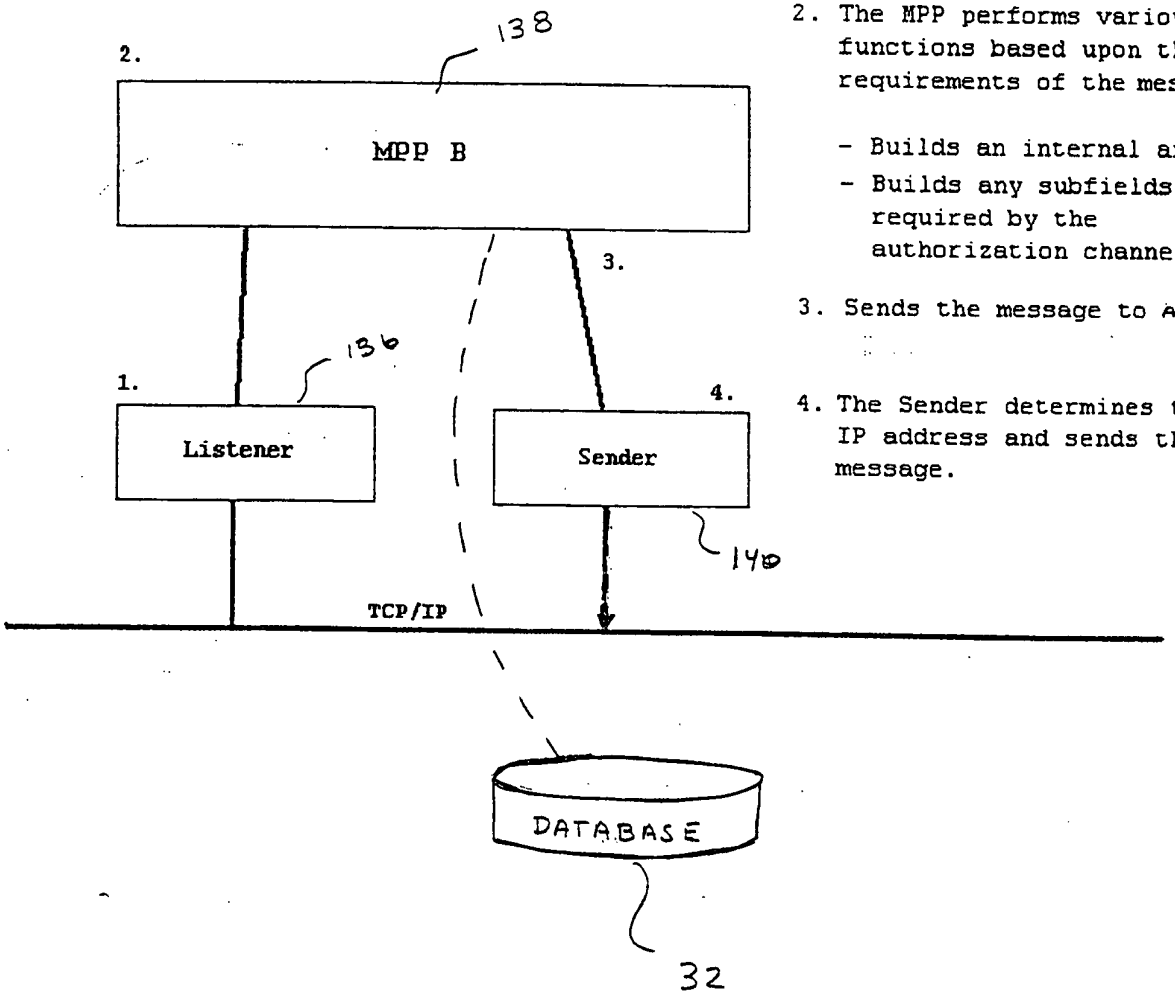
2. The MPP performs various functions based upon the requirements of the message.
   - Builds an internal array.
   - Parses composite fields into subfields of the array.
   - May perfrom authorization.
   - Determines who to send the message to. May be an MPP or MGR.
   - Builds a new message.

3. Sends a copy of the data to the database for archive.

4. Sends the message to the authorization host.

5. The Sender determines the IP address and sends the message.

FIG 5



1. The Listener buffers the data, then places the data onto the input queue of the MPP.

2. The MPP performs various functions based upon the requirements of the message.

   - Builds an internal array.
   - Builds any subfields required by the authorization channel.

3. Sends the message to AN MPP OR MGR.

4. The Sender determines the IP address and sends the message.

FIG 6

1. The listener buffers the message.

2. The MGR parses the message from ISO 8583 to the required format of the device/service.

3. The Device Driver removes the header and sends the message to the authorizor.

4. The Authorizor returns either an authorization or a denial for the transaction.

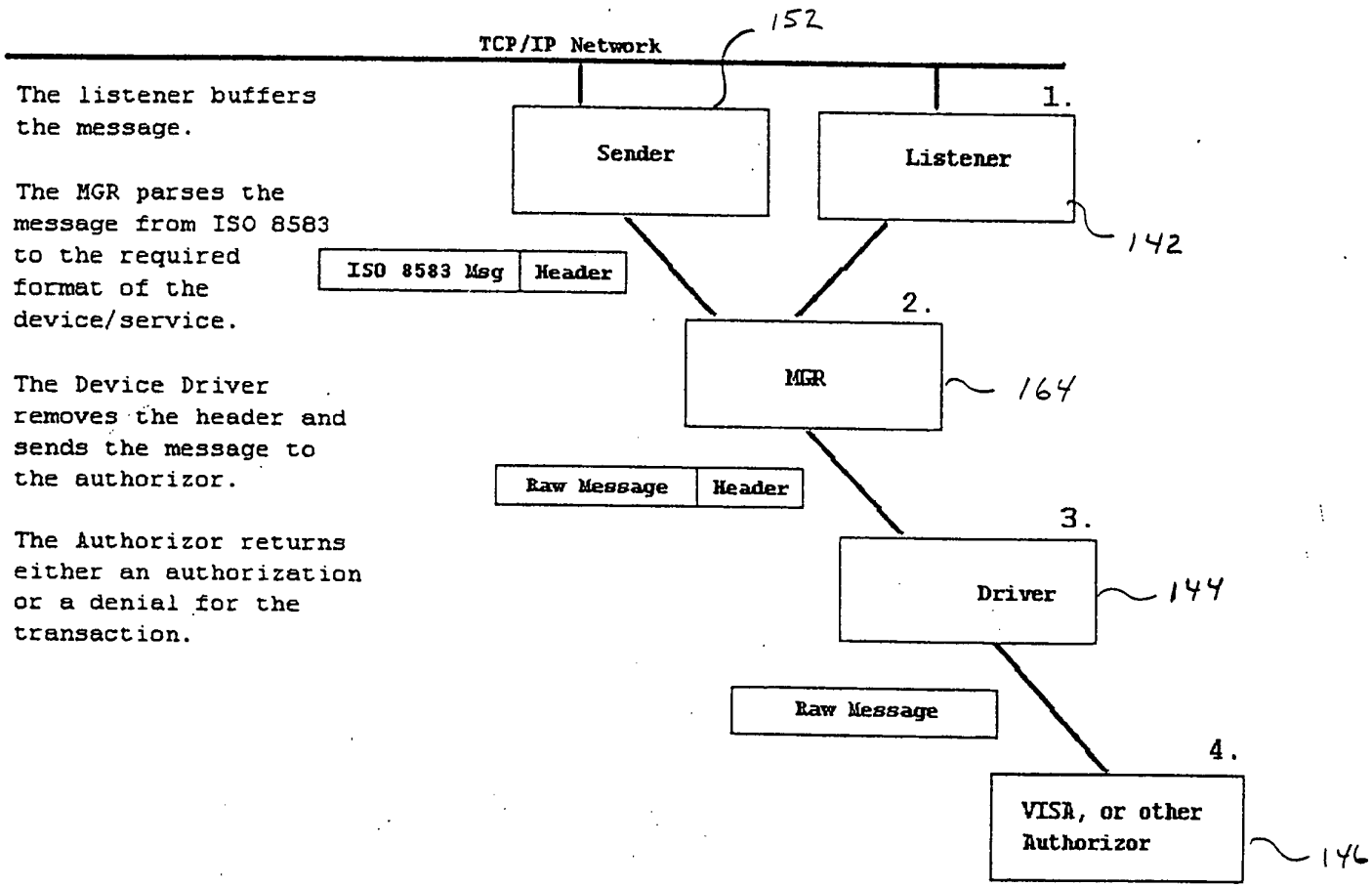


FIG 7

4. The Sender determines the IP address and sends the message.

3. The MGR parses the message from the RAW format to the ISO 8583 message format.

2. The device driver add a header and fills some fields after getting the information directly from the network or host.

1. The authorizor (VISA or other host) returns the message. This represents the actual host/network.

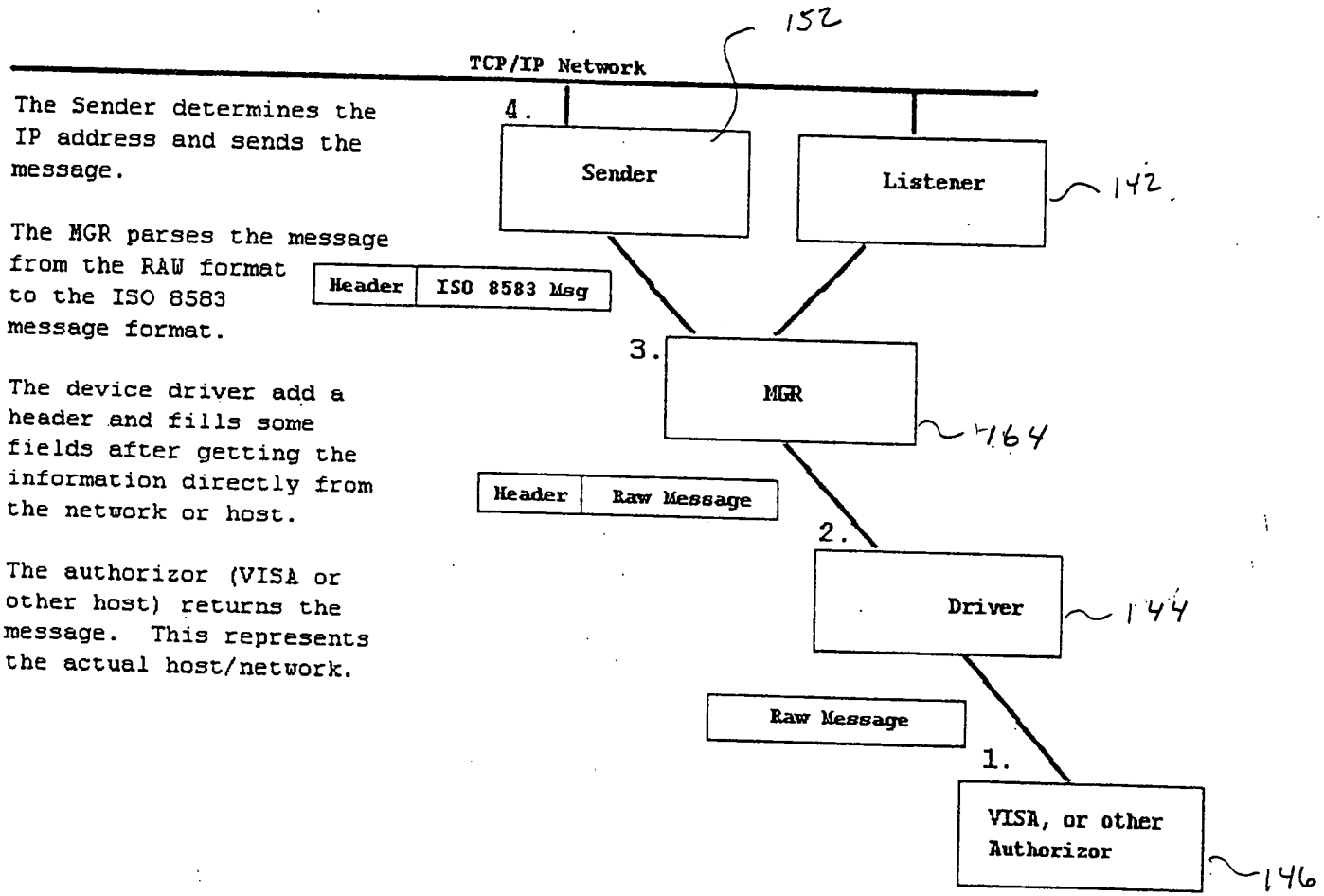
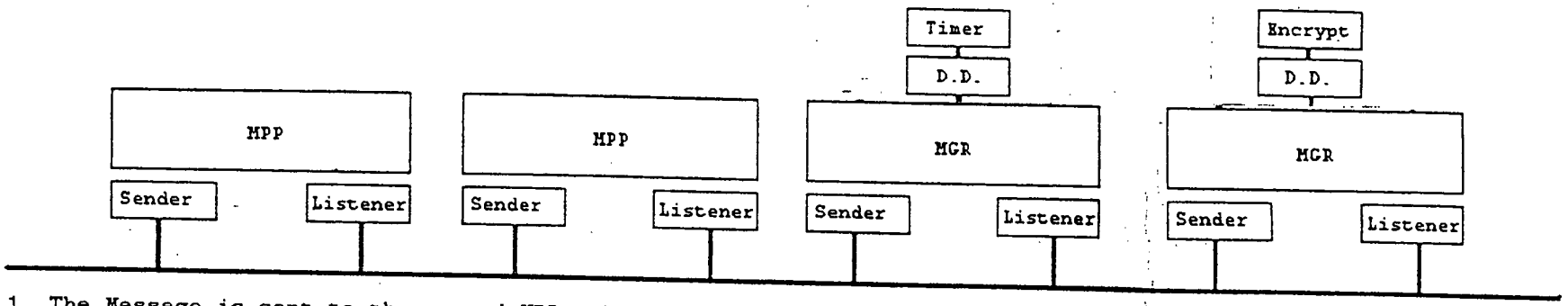
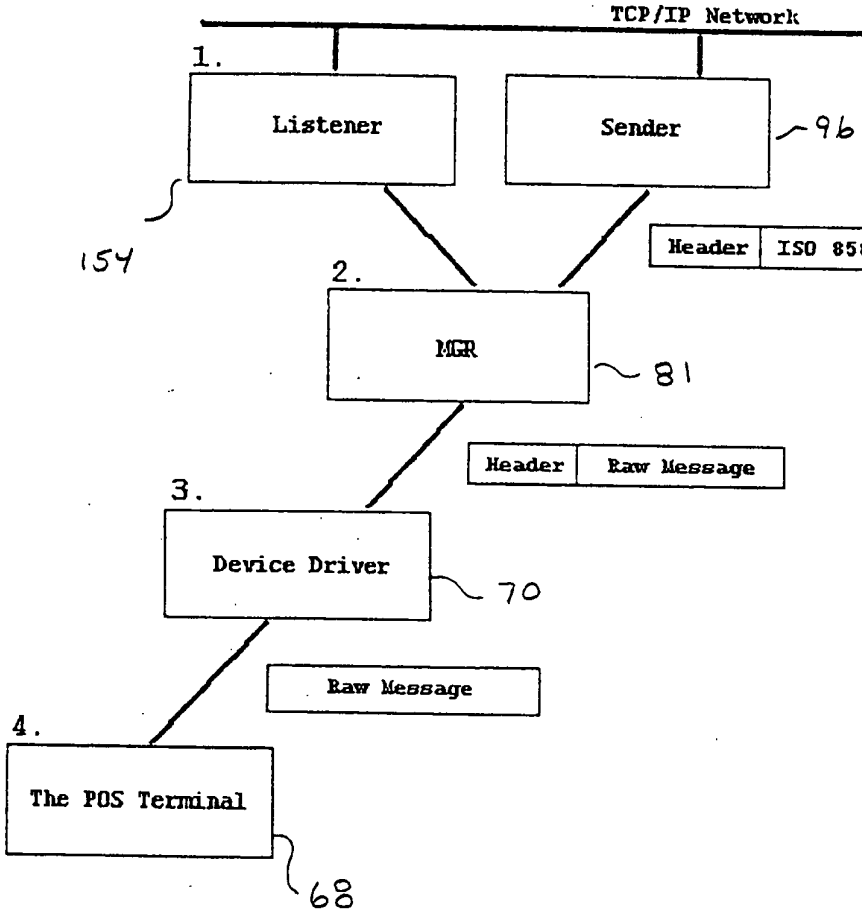


FIG 8



1. The Message is sent to the second MPP. It uses a echo-back field to determine the orgin of the message. The database contains the original message with a key. It may send the message to the first MPP by calling the Encryption Device for decryption of the PAN.
2. The message is received by the first MPP. It may need to build special fields, such as track II data. It will then send the message back to the original calling device by using the saved data in the database.

Fig 9



1. The Listener buffers the information.

2. The MRG parses the ISO 8583 message into a message format that can be used by the device.

3. The Device driver sends the message to the device.

4. The POS terminal returns a message confirming the authorization message.

The message is then returned to the MPP in the same manner as before.

FIG 10

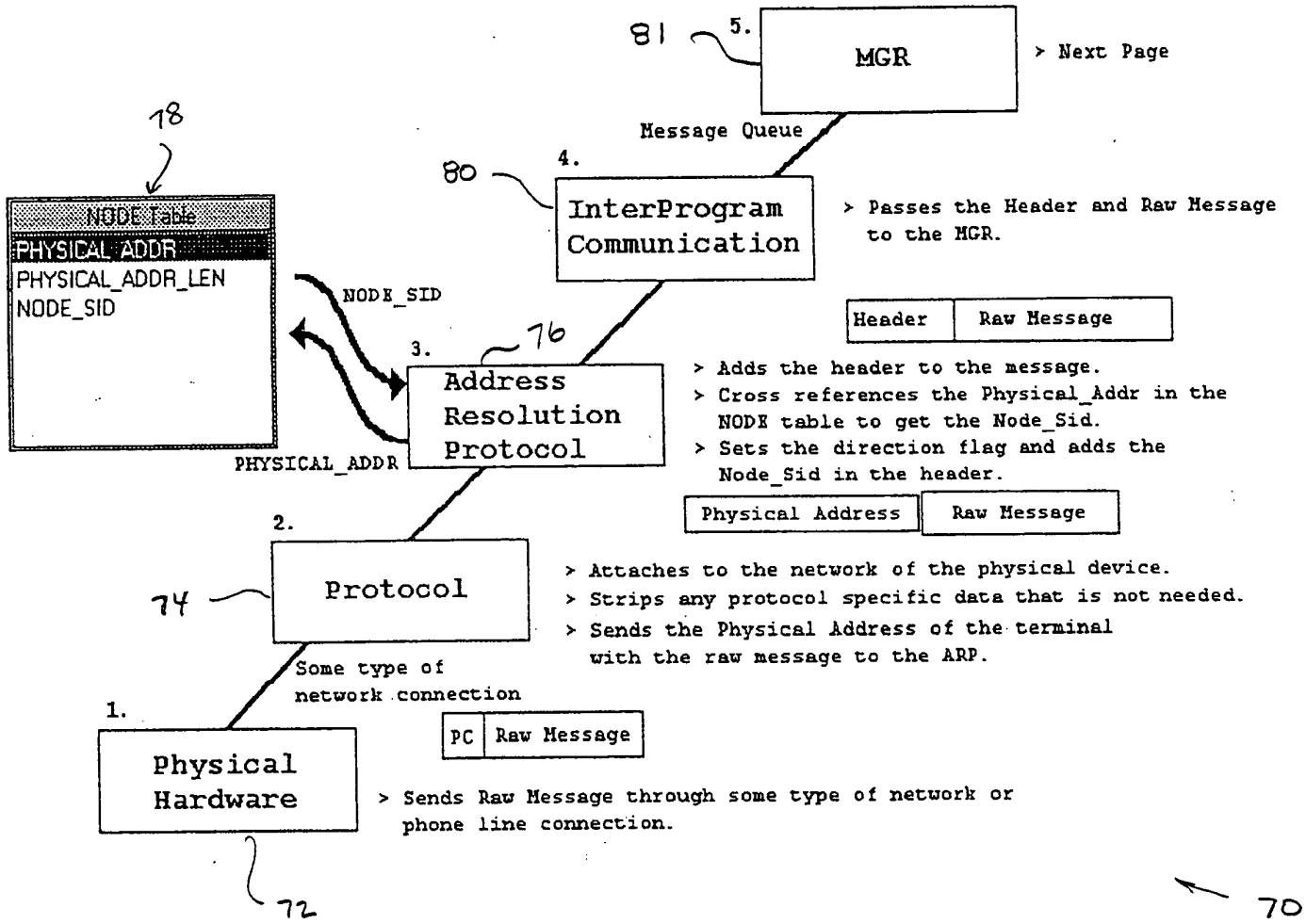

81
5.
MGR
> Next Page
Message Queue
78
80
4.
InterProgram Communication
> Passes the Header and Raw Message to the MGR.
NODE Table
PHYSICAL_ADDR
PHYSICAL_ADDR_LEN
NODE_SID
NODE_SID
PHYSICAL_ADDR
Header
Raw Message
3.
76
Address Resolution Protocol
> Adds the header to the message.
> Cross references the Physical_Addr in the NODE table to get the Node_Sid.
> Sets the direction flag and adds the Node_Sid in the header.
Physical Address
Raw Message
2.
74
Protocol
> Attaches to the network of the physical device.
> Strips any protocol specific data that is not needed.
> Sends the Physical Address of the terminal with the raw message to the ARP.
Some type of network connection
1.
PC
Raw Message
Physical Hardware
> Sends Raw Message through some type of network or phone line connection.
72
70

Fig 11

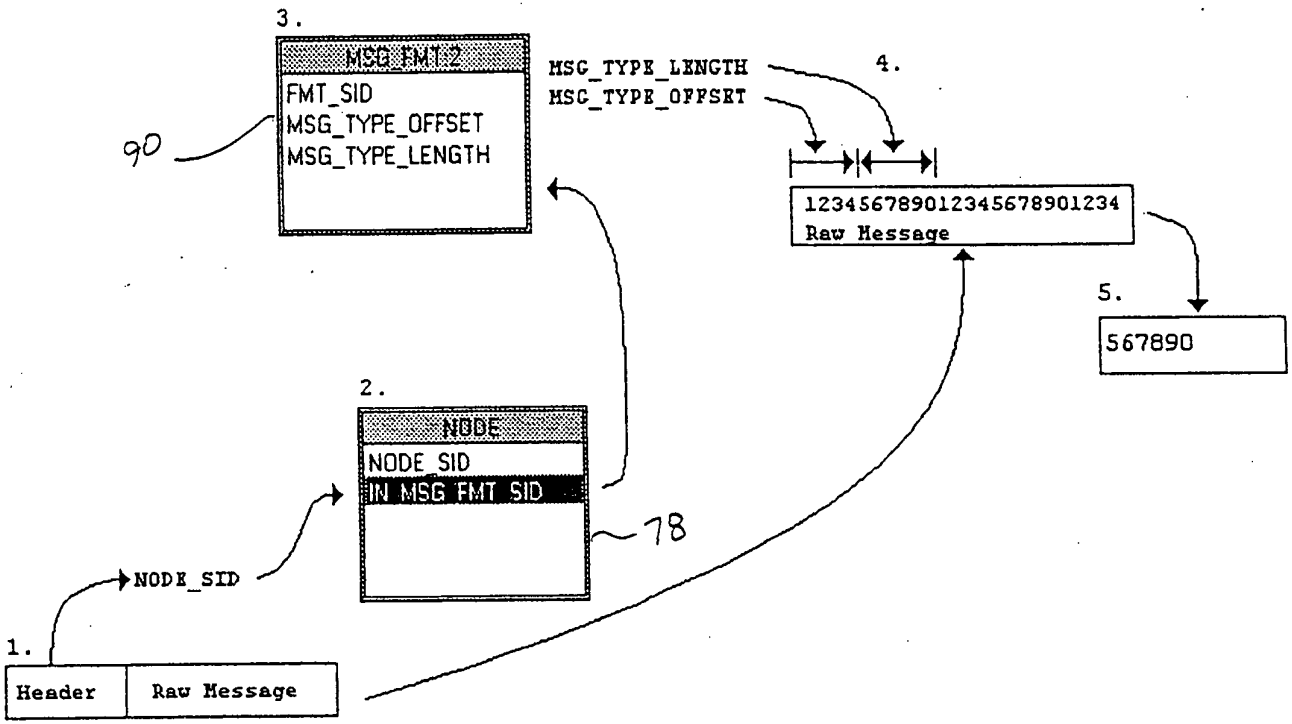

3.
MSG FMT 2
FMT_SID
MSG_TYPE_OFFSET
MSG_TYPE_LENGTH
90
MSG_TYPE_LENGTH
MSG_TYPE_OFFSET
4.
12345678901234567890123 4
Raw Message
5.
567890
2.
NODE
NODE_SID
IN MSG FMT SID
78
NODE_SID
1.
Header
Raw Message

FIG 12

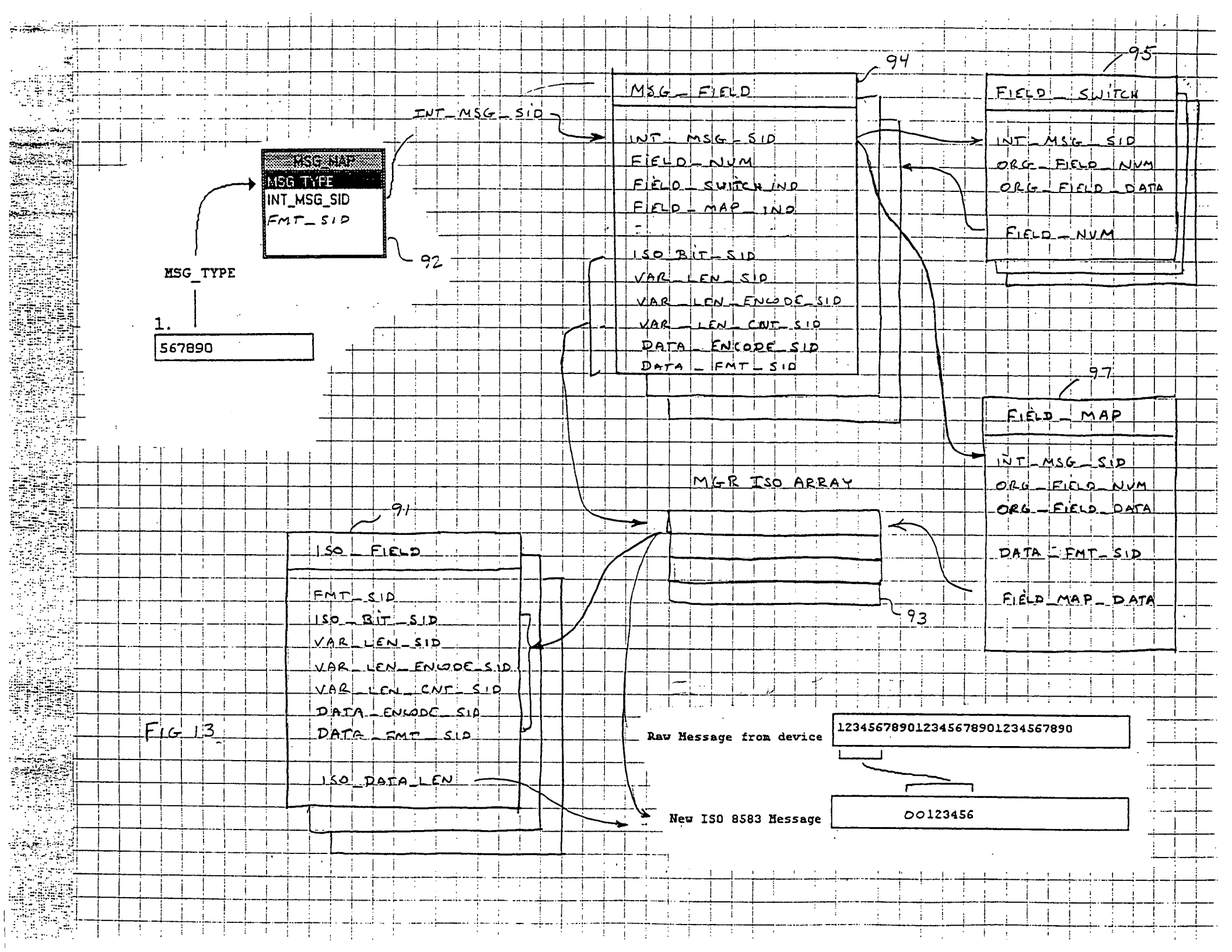

MSG_MAP
MSG TYPE
INT_MSG_SID
FMT_SID
92
MSG_TYPE
1.
567890
INT_MSG_SID
MSG_FIELD
94
INT_MSG_SID
FIELD_NUM
FIELD_SWITCH_IND
FIELD_MAP_IND
ISO_BIT_SID
VAR_LEN_SID
VAR_LEN_ENCODE_SID
VAR_LEN_CNT_SID
DATA_ENCODE_SID
DATA_FMT_SID
FIELD_SWITCH
95
INT_MSG_SID
ORG_FIELD_NUM
ORG_FIELD_DATA
FIELD_NUM
FIELD_MAP
97
INT_MSG_SID
ORG_FIELD_NUM
ORG_FIELD_DATA
DATA_FMT_SID
FIELD_MAP_DATA
MGR ISO ARRAY
93
ISO_FIELD
91
FMT_SID
ISO_BIT_SID
VAR_LEN_SID
VAR_LEN_ENCODE_SID
VAR_LEN_CNT_SID
DATA_ENCODE_SID
DATA_FMT_SID
ISO_DATA_LEN
Raw Message from device
12345678901234567890123456789O
New ISO 8583 Message
00123456

FIG 13

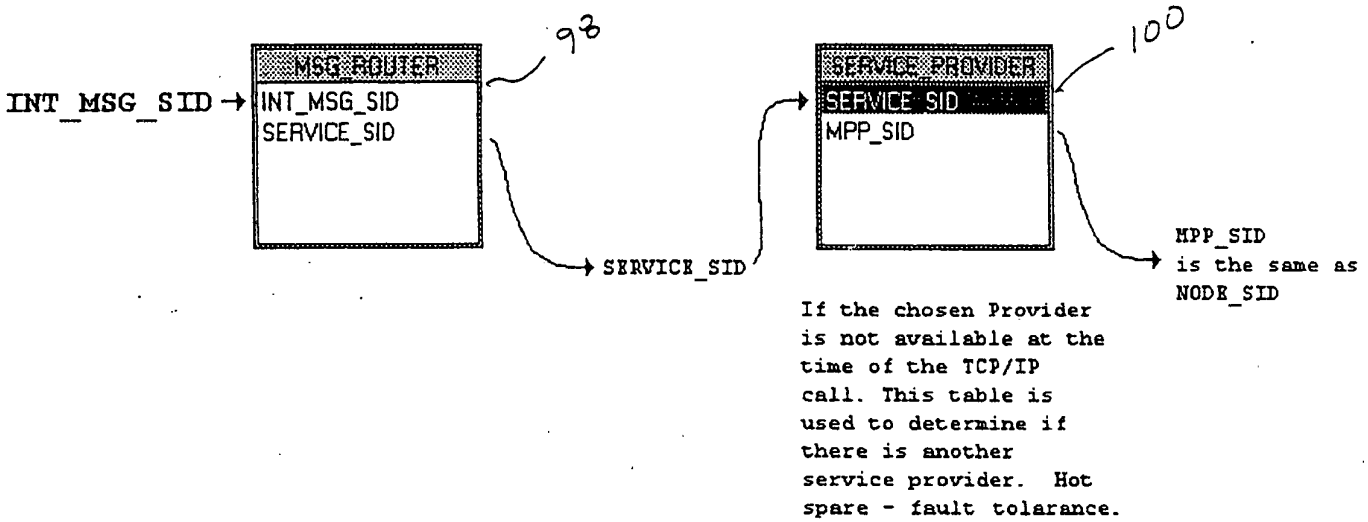

INT_MSG_SID
MSG ROUTER
INT_MSG_SID
SERVICE_SID
98
SERVICE_SID
SERVICE PROVIDER
SERVICE_SID
MPP_SID
100
MPP_SID
is the same as
NODE_SID
If the chosen Provider
is not available at the
time of the TCP/IP
call. This table is
used to determine if
there is another
service provider. Hot
spare - fault tolarance.

Fig 14

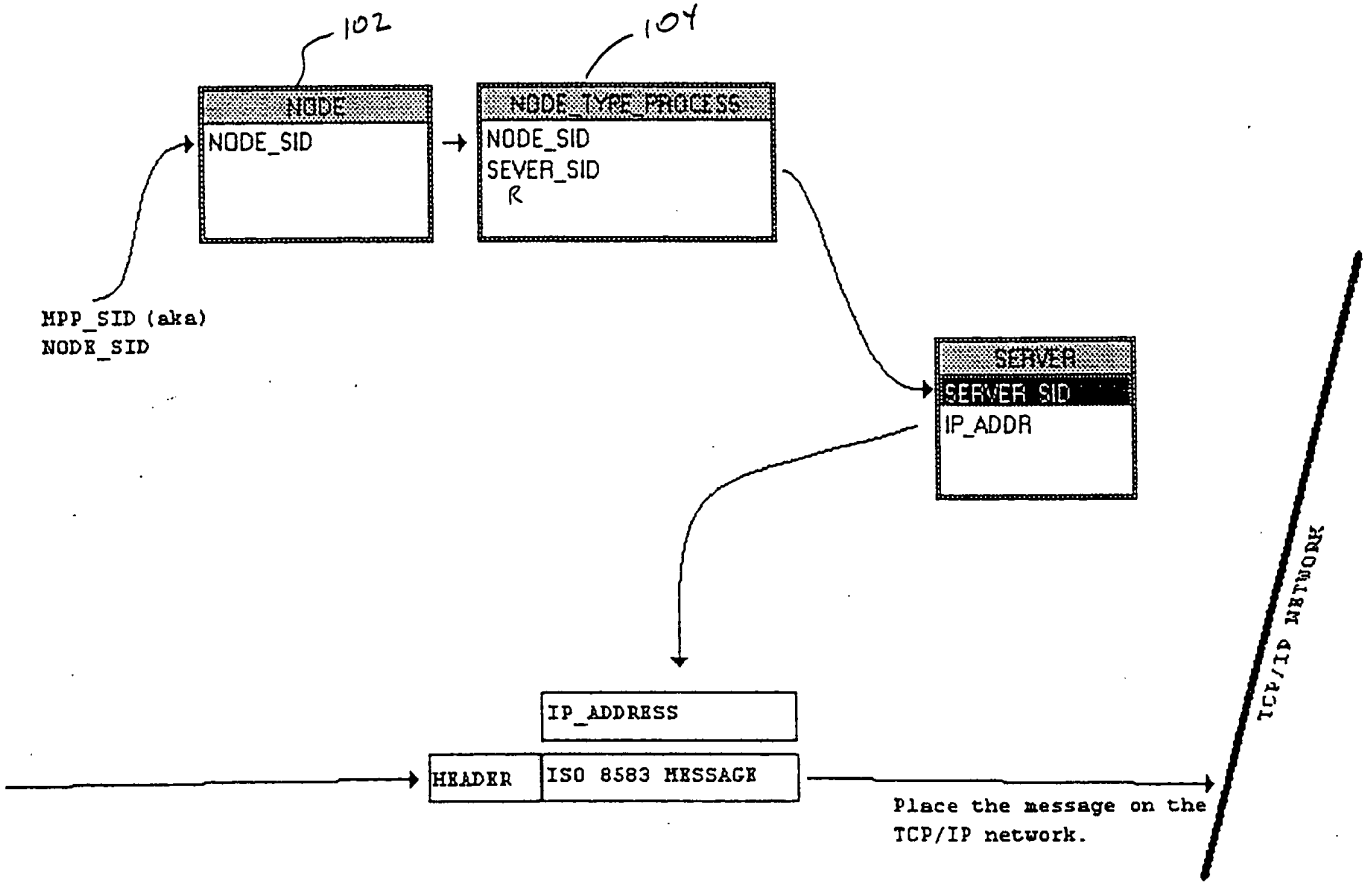

102
104
NODE
NODE_SID
NODE_TYPE_PROCESS
NODE_SID
SEVER_SID
R
MPP_SID (aka)
NODE_SID
SERVER
SERVER_SID
IP_ADDR
IP_ADDRESS
HEADER
ISO 8583 MESSAGE
Place the message on the
TCP/IP network.
TCP/IP NETWORK

Fig 15

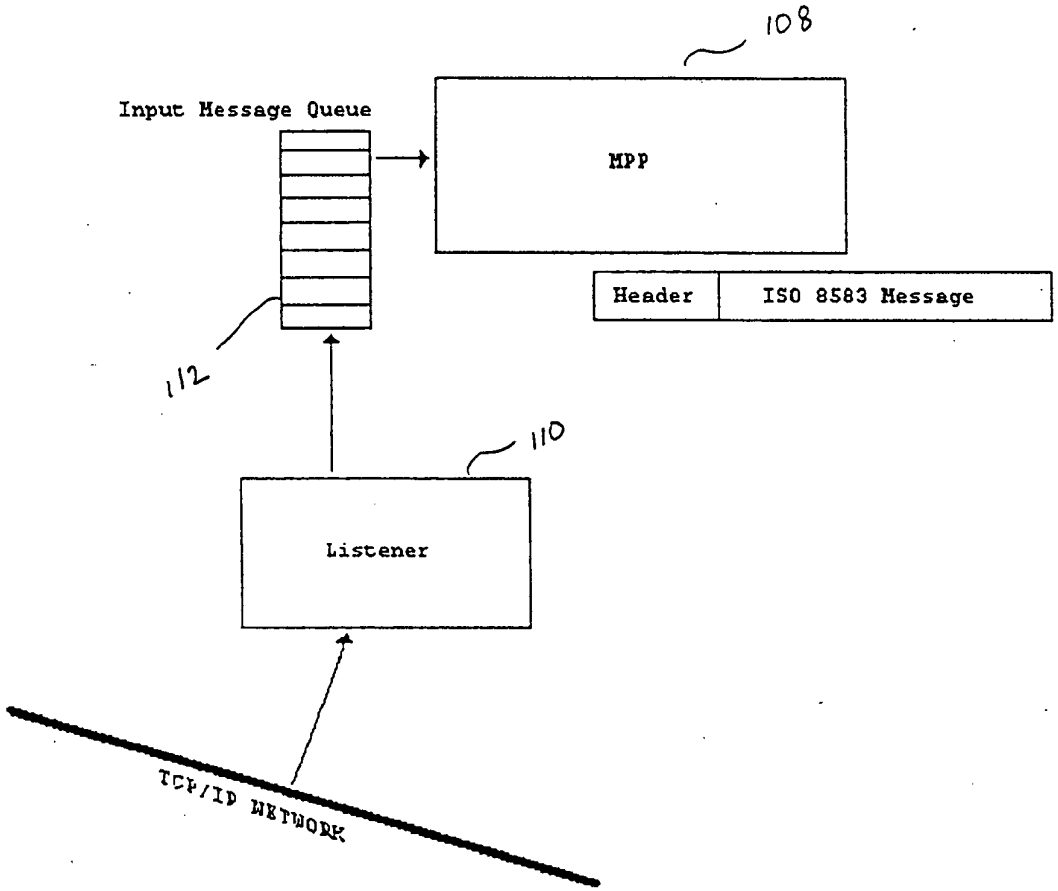

108
Input Message Queue
MPP
Header
ISO 8583 Message
112
110
Listener
TCP/IP NETWORK

Fig 16

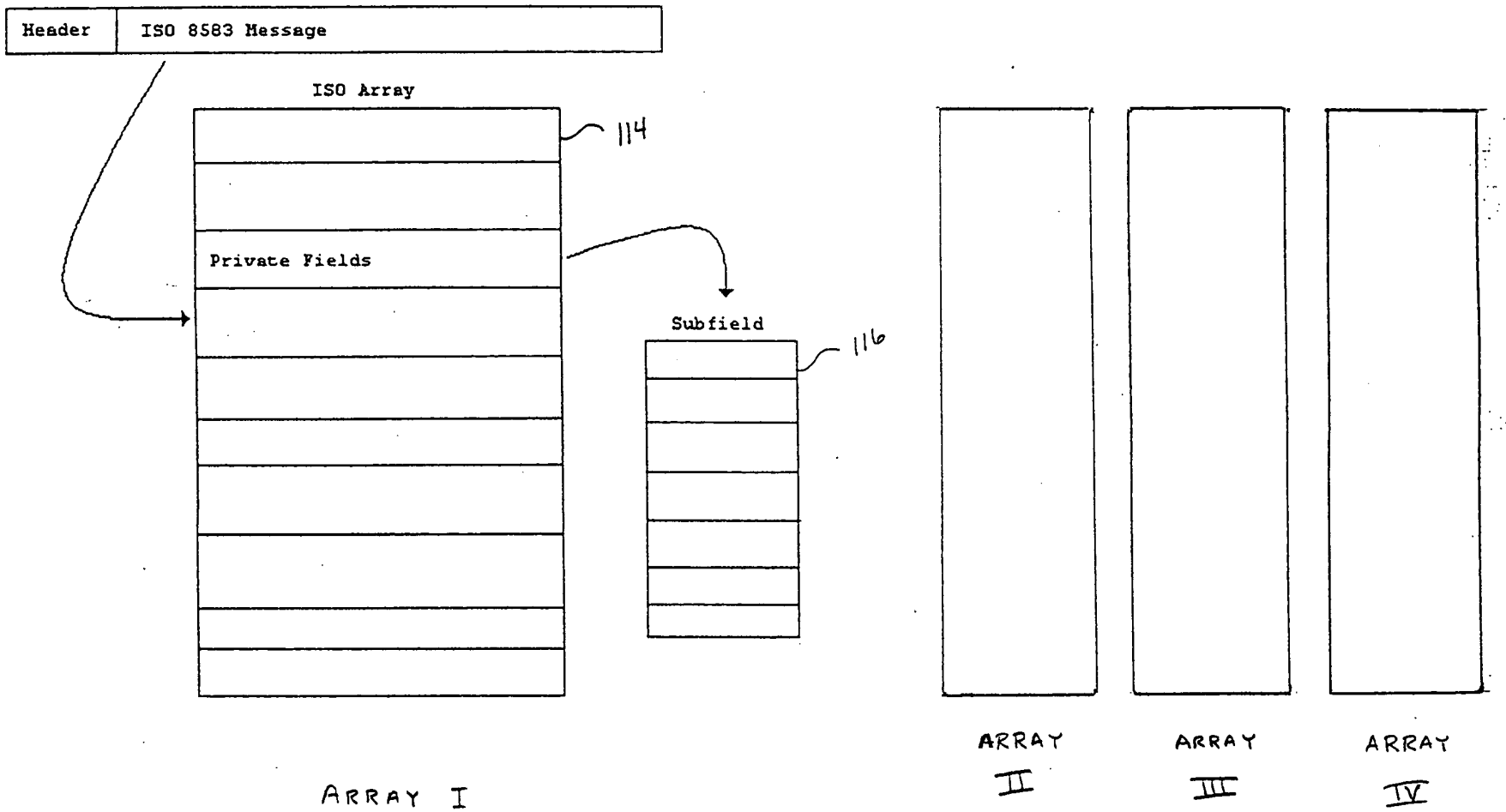

Header
ISO 8583 Message
ISO Array
114
Private Fields
Subfield
116
ARRAY I
ARRAY II
ARRAY III
ARRAY IV

Fig 17

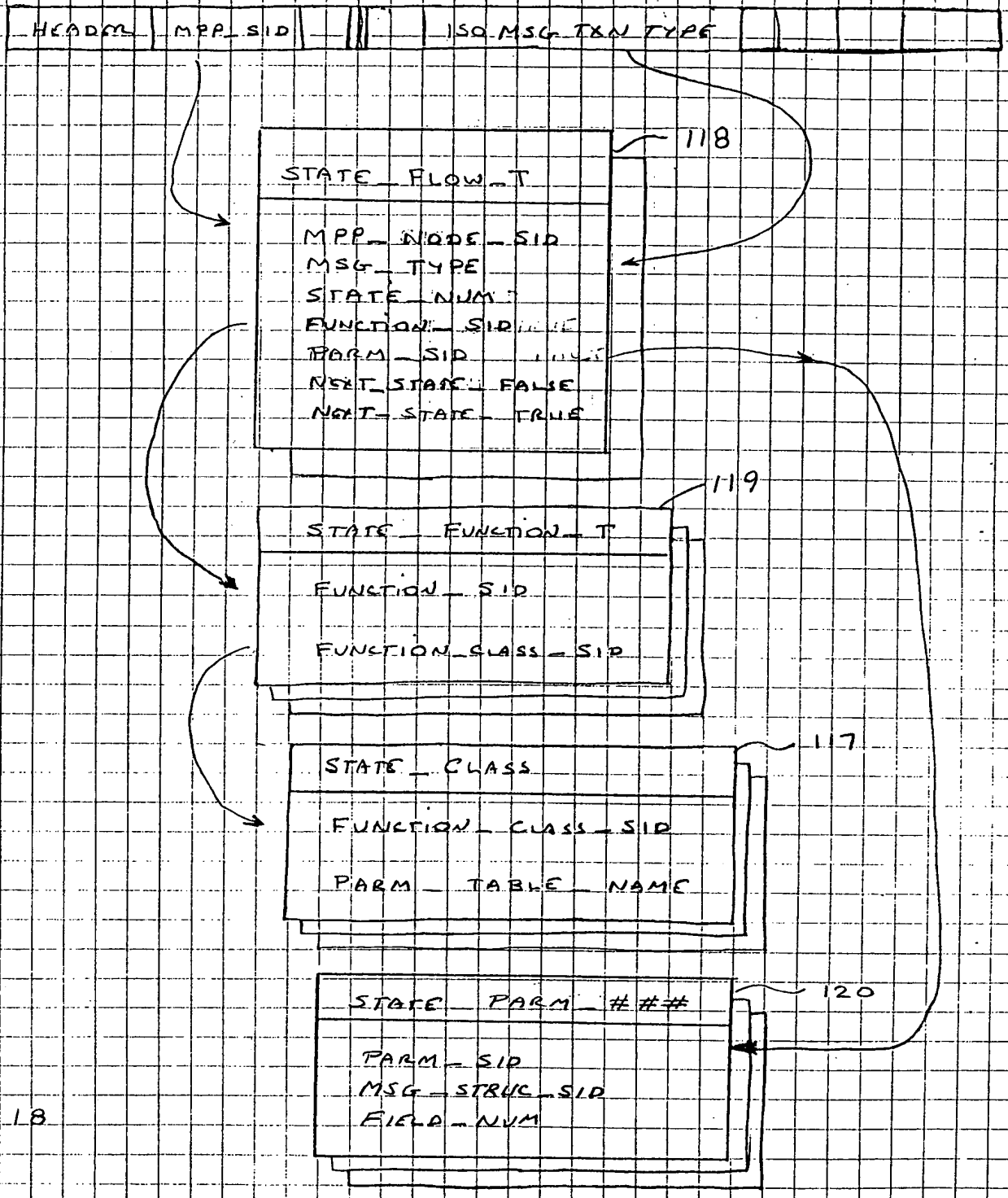

HEADER
MPP_SID
ISO_MSG_TXN_TYPE
118
STATE_FLOW_T
MPP_NODE_SID
MSG_TYPE
STATE_NUM
FUNCTION_SID
PARM_SID
NEXT_STATE_FALSE
NEXT_STATE_TRUE
119
STATE_FUNCTION_T
FUNCTION_SID
FUNCTION_CLASS_SID
117
STATE_CLASS
FUNCTION_CLASS_SID
PARM_TABLE_NAME
120
STATE_PARM_###
PARM_SID
MSG_STRUC_SID
FIELD_NUM

FIG. 18

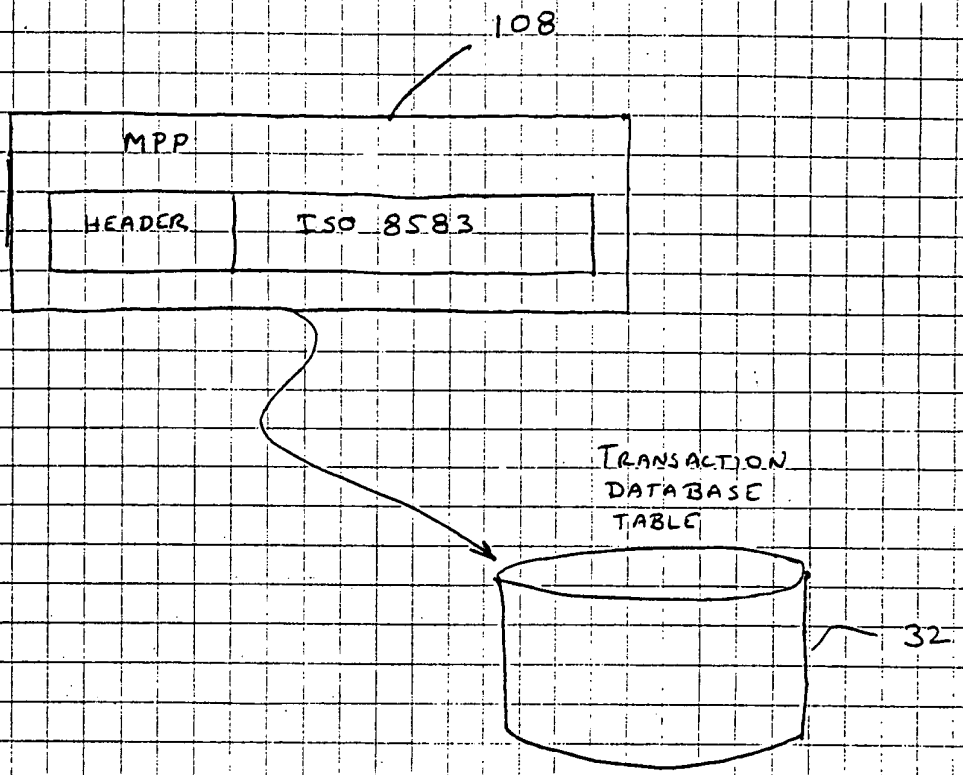


FIGURE 19

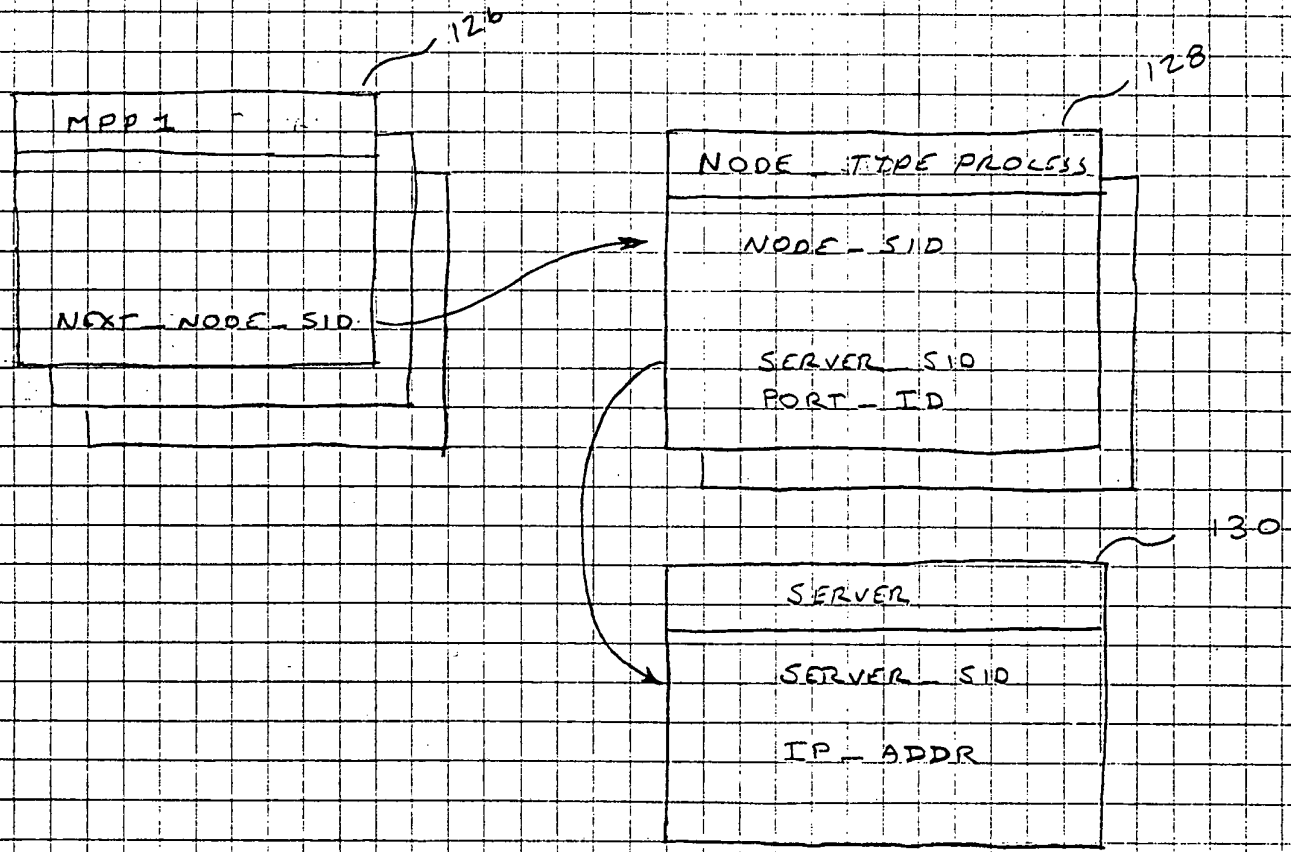
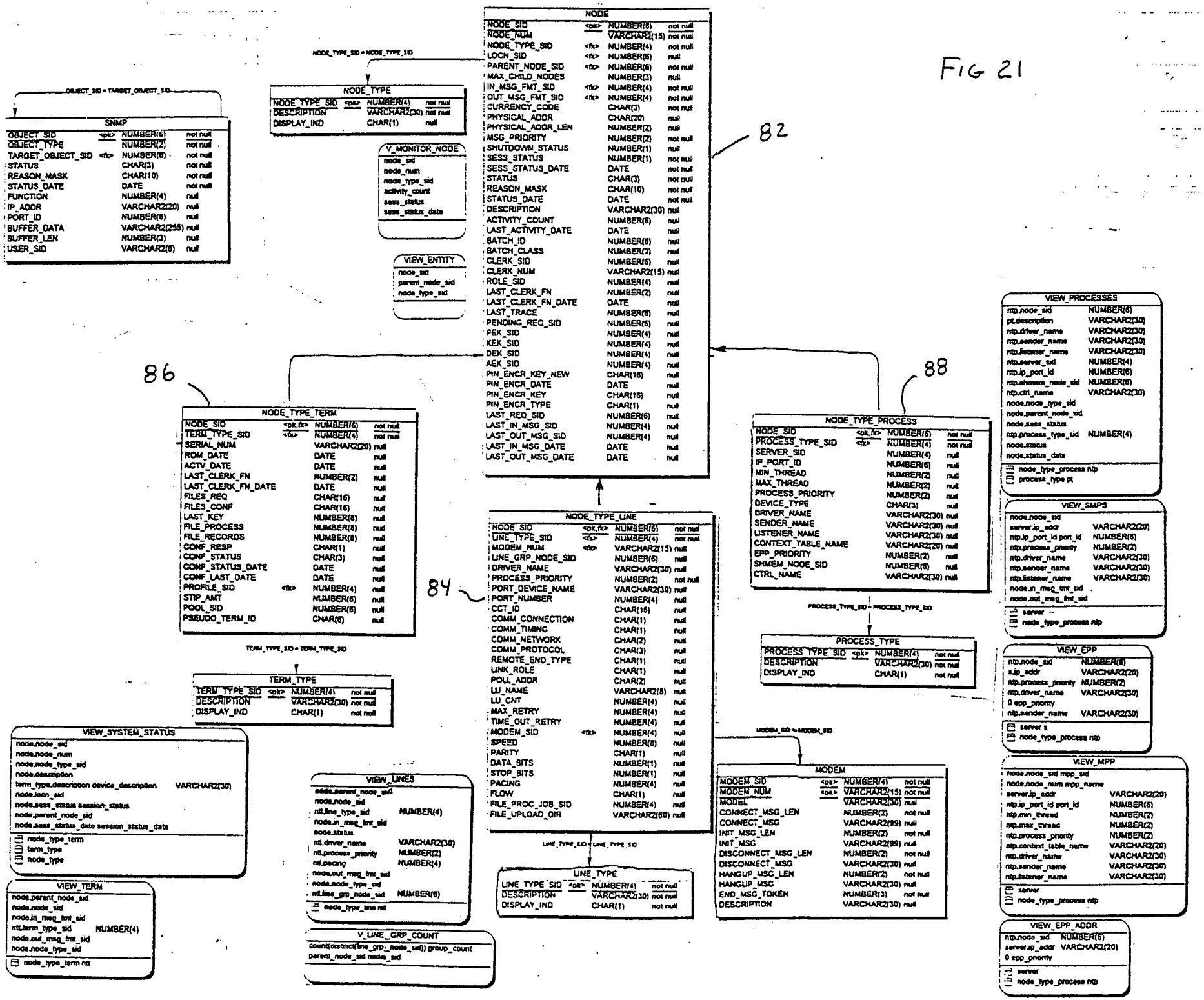

126
MPP 1
NEXT_NODE_SID
128
NODE_TYPE PROCESS
NODE_SID
SERVER_SID
PORT_ID
130
SERVER
SERVER_SID
IP_ADDR

FIG 20

FIG 21


NODE
NODE_SID <pk> NUMBER(6) not null
NODE_NUM VARCHAR2(15) not null
NODE_TYPE_SID <fk> NUMBER(4) not null
LOCN_SID <fk> NUMBER(6) null
PARENT_NODE_SID <fk> NUMBER(6) not null
MAX_CHILD_NODES NUMBER(3) null
IN_MSG_FMT_SID <fk> NUMBER(4) not null
OUT_MSG_FMT_SID <fk> NUMBER(4) not null
CURRENCY_CODE CHAR(3) not null
PHYSICAL_ADDR CHAR(20) null
PHYSICAL_ADDR_LEN NUMBER(2) null
MSG_PRIORITY NUMBER(2) not null
SHUTDOWN_STATUS NUMBER(1) null
SESS_STATUS NUMBER(1) not null
SESS_STATUS_DATE DATE not null
STATUS CHAR(3) not null
REASON_MASK CHAR(10) not null
STATUS_DATE DATE not null
DESCRIPTION VARCHAR2(30) null
ACTIVITY_COUNT NUMBER(6) null
LAST_ACTIVITY_DATE DATE null
BATCH_ID NUMBER(8) null
BATCH_CLASS NUMBER(3) null
CLERK_SID NUMBER(6) null
CLERK_NUM VARCHAR2(15) null
ROLE_SID NUMBER(4) null
LAST_CLERK_FN NUMBER(2) null
LAST_CLERK_FN_DATE DATE null
LAST_TRACE NUMBER(6) null
PENDING_REQ_SID NUMBER(6) null
PEX_SID NUMBER(4) null
KEK_SID NUMBER(4) null
DEK_SID NUMBER(4) null
AEK_SID NUMBER(4) null
PIN_ENCR_KEY_NEW CHAR(16) null
PIN_ENCR_DATE DATE null
PIN_ENCR_KEY CHAR(16) null
PIN_ENCR_TYPE CHAR(1) null
LAST_REQ_SID NUMBER(6) null
LAST_IN_MSG_SID NUMBER(4) null
LAST_OUT_MSG_SID NUMBER(4) null
LAST_IN_MSG_DATE DATE null
LAST_OUT_MSG_DATE DATE null
82
NODE_TYPE_SID = NODE_TYPE_SID
NODE_TYPE
NODE_TYPE_SID <pk> NUMBER(4) not null
DESCRIPTION VARCHAR2(30) not null
DISPLAY_IND CHAR(1) null
OBJECT_SID = TARGET_OBJECT_SID
SNMP
OBJECT_SID <pk> NUMBER(6) not null
OBJECT_TYPE NUMBER(2) not null
TARGET_OBJECT_SID <fk> NUMBER(6) not null
STATUS CHAR(3) not null
REASON_MASK CHAR(10) not null
STATUS_DATE DATE not null
FUNCTION NUMBER(4) null
IP_ADDR VARCHAR2(20) null
PORT_ID NUMBER(8) null
BUFFER_DATA VARCHAR2(255) null
BUFFER_LEN NUMBER(3) null
USER_SID VARCHAR2(6) null
V_MONITOR_NODE
node_sid
node_num
node_type_sid
activity_count
sess_status
sess_status_date
VIEW_ENTITY
node_sid
parent_node_sid
node_type_sid
86
NODE_TYPE_TERM
NODE_SID <pk,fk> NUMBER(6) not null
TERM_TYPE_SID <fk> NUMBER(4) not null
SERIAL_NUM VARCHAR2(20) null
ROM_DATE DATE null
ACTV_DATE DATE null
LAST_CLERK_FN NUMBER(2) null
LAST_CLERK_FN_DATE DATE null
FILES_REQ CHAR(16) null
FILES_CONF CHAR(16) null
LAST_KEY NUMBER(8) null
FILE_PROCESS NUMBER(8) null
FILE_RECORDS NUMBER(8) null
CONF_RESP CHAR(1) null
CONF_STATUS CHAR(3) null
CONF_STATUS_DATE DATE null
CONF_LAST_DATE DATE null
PROFILE_SID <fk> NUMBER(4) null
STIP_AMT NUMBER(6) null
POOL_SID NUMBER(5) null
PSEUDO_TERM_ID CHAR(6) null
TERM_TYPE_SID = TERM_TYPE_SID
TERM_TYPE
TERM_TYPE_SID <pk> NUMBER(4) not null
DESCRIPTION VARCHAR2(30) not null
DISPLAY_IND CHAR(1) not null
84
NODE_TYPE_LINE
NODE_SID <pk,fk> NUMBER(6) not null
LINE_TYPE_SID <fk> NUMBER(4) not null
MODEM_NUM <fk> VARCHAR2(15) null
LINE_GRP_NODE_SID NUMBER(6) null
DRIVER_NAME VARCHAR2(30) null
PROCESS_PRIORITY NUMBER(2) not null
PORT_DEVICE_NAME VARCHAR2(30) null
PORT_NUMBER NUMBER(4) null
CCT_ID CHAR(16) null
COMM_CONNECTION CHAR(1) null
COMM_TIMING CHAR(1) null
COMM_NETWORK CHAR(2) null
COMM_PROTOCOL CHAR(3) null
REMOTE_END_TYPE CHAR(1) null
LINK_ROLE CHAR(1) null
POLL_ADDR CHAR(2) null
LU_NAME VARCHAR2(8) null
LU_CNT NUMBER(4) null
MAX_RETRY NUMBER(4) null
TIME_OUT_RETRY NUMBER(4) null
MODEM_SID <fk> NUMBER(4) null
SPEED NUMBER(8) null
PARITY CHAR(1) null
DATA_BITS NUMBER(1) null
STOP_BITS NUMBER(1) null
PACING NUMBER(4) null
FLOW CHAR(1) null
FILE_PROC_JOB_SID NUMBER(4) null
FILE_UPLOAD_DIR VARCHAR2(60) null
LINE_TYPE_SID = LINE_TYPE_SID
LINE_TYPE
LINE_TYPE_SID <pk> NUMBER(4) not null
DESCRIPTION VARCHAR2(30) not null
DISPLAY_IND CHAR(1) not null
MODEM_SID = MODEM_SID
MODEM
MODEM_SID <pk> NUMBER(4) not null
MODEM_NUM <pak> VARCHAR2(15) not null
MODEL VARCHAR2(30) null
CONNECT_MSG_LEN NUMBER(2) not null
CONNECT_MSG VARCHAR2(99) null
INIT_MSG_LEN NUMBER(2) not null
INIT_MSG VARCHAR2(99) null
DISCONNECT_MSG_LEN NUMBER(2) not null
DISCONNECT_MSG VARCHAR2(30) null
HANGUP_MSG_LEN NUMBER(2) not null
HANGUP_MSG VARCHAR2(30) null
END_MSG_TOKEN NUMBER(3) not null
DESCRIPTION VARCHAR2(30) null
88
NODE_TYPE_PROCESS
NODE_SID <pk,fk> NUMBER(6) not null
PROCESS_TYPE_SID <fk> NUMBER(4) not null
SERVER_SID NUMBER(4) null
IP_PORT_ID NUMBER(6) null
MIN_THREAD NUMBER(2) null
MAX_THREAD NUMBER(2) null
PROCESS_PRIORITY NUMBER(2) null
DEVICE_TYPE CHAR(3) null
DRIVER_NAME VARCHAR2(30) null
SENDER_NAME VARCHAR2(30) null
LISTENER_NAME VARCHAR2(30) null
CONTEXT_TABLE_NAME VARCHAR2(20) null
EPP_PRIORITY NUMBER(2) null
SHMEM_NODE_SID NUMBER(6) null
CTRL_NAME VARCHAR2(30) null
PROCESS_TYPE_SID = PROCESS_TYPE_SID
PROCESS_TYPE
PROCESS_TYPE_SID <pk> NUMBER(4) not null
DESCRIPTION VARCHAR2(30) not null
DISPLAY_IND CHAR(1) not null
VIEW_PROCESSES
ntp.node_sid NUMBER(6)
pt.description VARCHAR2(30)
ntp.driver_name VARCHAR2(30)
ntp.sender_name VARCHAR2(30)
ntp.listener_name VARCHAR2(30)
ntp.server_sid NUMBER(4)
ntp.ip_port_id NUMBER(6)
ntp.shmem_node_sid NUMBER(6)
ntp.ctrl_name VARCHAR2(30)
node.node_type_sid
node.parent_node_sid
node.sess_status
ntp.process_type_sid NUMBER(4)
node.status
node.status_date
node_type_process ntp
process_type pt
VIEW_SMPS
node.node_sid
server.ip_addr VARCHAR2(20)
ntp.ip_port_id port_id NUMBER(6)
ntp.process_priority NUMBER(2)
ntp.driver_name VARCHAR2(30)
ntp.sender_name VARCHAR2(30)
ntp.listener_name VARCHAR2(30)
node.in_msg_fmt_sid
node.out_msg_fmt_sid
server --
node_type_process ntp
VIEW_EPP
ntp.node_sid NUMBER(6)
s.ip_addr VARCHAR2(20)
ntp.process_priority NUMBER(2)
ntp.driver_name VARCHAR2(30)
0 epp_priority
ntp.sender_name VARCHAR2(30)
server s
node_type_process ntp
VIEW_MPP
node.node_sid mpp_sid
node.node_num mpp_name
server.ip_addr VARCHAR2(20)
ntp.ip_port_id port_id NUMBER(6)
ntp.min_thread NUMBER(2)
ntp.max_thread NUMBER(2)
ntp.process_priority NUMBER(2)
ntp.context_table_name VARCHAR2(20)
ntp.driver_name VARCHAR2(30)
ntp.sender_name VARCHAR2(30)
ntp.listener_name VARCHAR2(30)
server
node_type_process ntp
VIEW_EPP_ADDR
ntp.node_sid NUMBER(6)
server.ip_addr VARCHAR2(20)
0 epp_priority
server
node_type_process ntp
VIEW_SYSTEM_STATUS
node.node_sid
node.node_num
node.node_type_sid
node.description
term_type.description device_description VARCHAR2(30)
node.locn_sid
node.sess_status session_status
node.parent_node_sid
node.sess_status_date session_status_date
node_type_term
term_type
node_type
VIEW_TERM
node.parent_node_sid
node.node_sid
node.in_msg_fmt_sid
ntt.term_type_sid NUMBER(4)
node.out_msg_fmt_sid
node.node_type_sid
node_type_term ntt
VIEW_LINES
node.parent_node_sid
node.node_sid
ntl.line_type_sid NUMBER(4)
node.in_msg_fmt_sid
node.status
ntl.driver_name VARCHAR2(30)
ntl.process_priority NUMBER(2)
ntl.pacing NUMBER(4)
node.out_msg_fmt_sid
node.node_type_sid
ntl.line_grp_node_sid NUMBER(6)
node_type_line ntl
V_LINE_GRP_COUNT
count(distinct(line_grp:_node_sid)) group_count
parent_node_sid node_sid

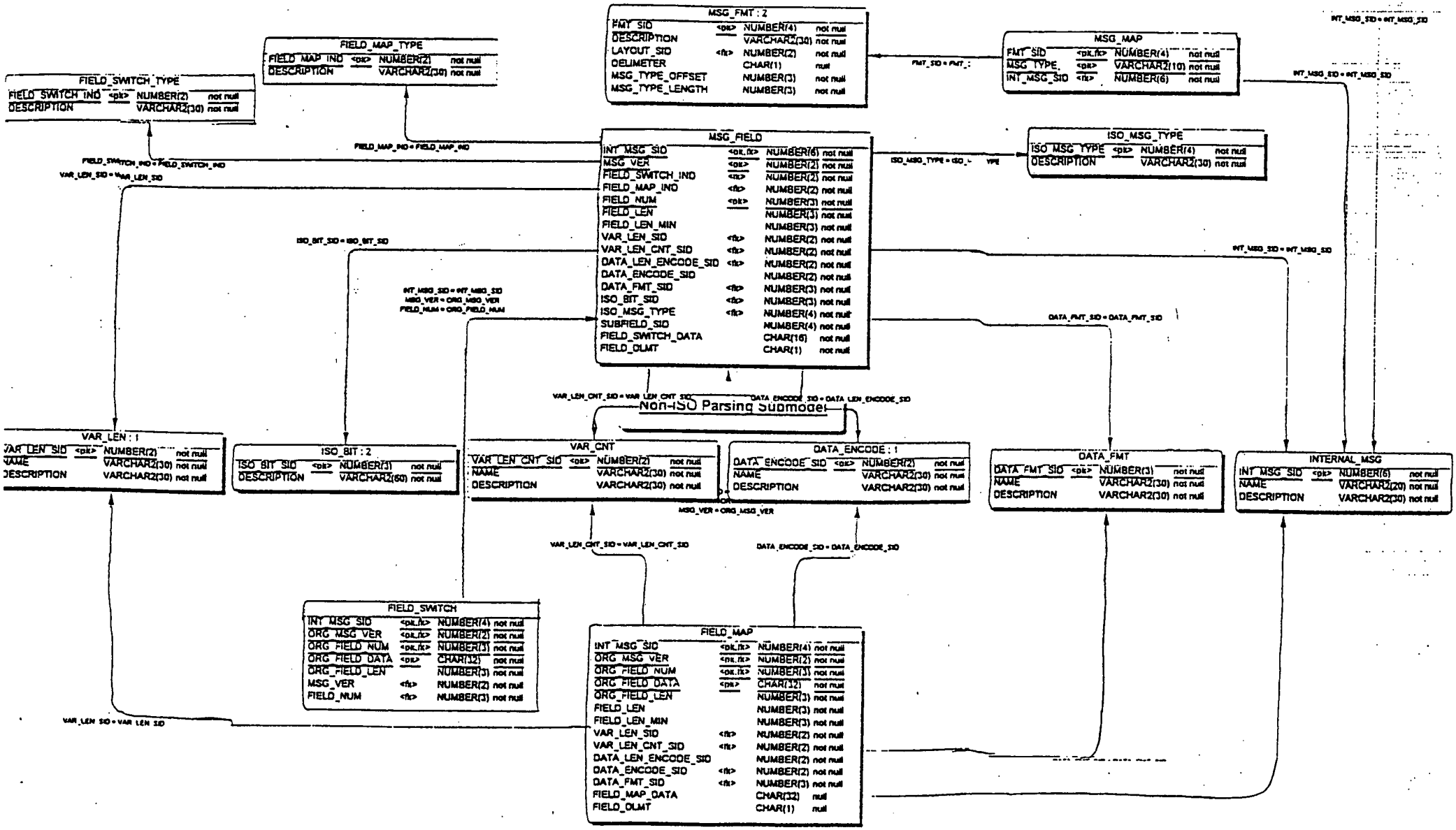

MSG_FMT : 2
FMT_SID <pk> NUMBER(4) not null
DESCRIPTION VARCHAR2(30) not null
LAYOUT_SID <fk> NUMBER(2) not null
DELIMETER CHAR(1) null
MSG_TYPE_OFFSET NUMBER(3) not null
MSG_TYPE_LENGTH NUMBER(3) not null
FIELD_MAP_TYPE
FIELD_MAP_IND <pk> NUMBER(2) not null
DESCRIPTION VARCHAR2(30) not null
MSG_MAP
FMT_SID <pk,fk> NUMBER(4) not null
MSG_TYPE <pk> VARCHAR2(10) not null
INT_MSG_SID <fk> NUMBER(6) not null
FIELD_SWITCH_TYPE
FIELD_SWITCH_IND <pk> NUMBER(2) not null
DESCRIPTION VARCHAR2(30) not null
MSG_FIELD
INT_MSG_SID <pk,fk> NUMBER(6) not null
MSG_VER <pk> NUMBER(2) not null
FIELD_SWITCH_IND <fk> NUMBER(2) not null
FIELD_MAP_IND <fk> NUMBER(2) not null
FIELD_NUM <pk> NUMBER(3) not null
FIELD_LEN NUMBER(3) not null
FIELD_LEN_MIN NUMBER(3) not null
VAR_LEN_SID <fk> NUMBER(2) not null
VAR_LEN_CNT_SID <fk> NUMBER(2) not null
DATA_LEN_ENCODE_SID <fk> NUMBER(2) not null
DATA_ENCODE_SID NUMBER(2) not null
DATA_FMT_SID <fk> NUMBER(3) not null
ISO_BIT_SID <fk> NUMBER(3) not null
ISO_MSG_TYPE <fk> NUMBER(4) not null
SUBFIELD_SID NUMBER(4) not null
FIELD_SWITCH_DATA CHAR(16) not null
FIELD_DLMT CHAR(1) not null
ISO_MSG_TYPE
ISO_MSG_TYPE <pk> NUMBER(4) not null
DESCRIPTION VARCHAR2(30) not null
Non-ISO Parsing Submodel
VAR_LEN : 1
VAR_LEN_SID <pk> NUMBER(2) not null
NAME VARCHAR2(30) not null
DESCRIPTION VARCHAR2(30) not null
ISO_BIT : 2
ISO_BIT_SID <pk> NUMBER(3) not null
DESCRIPTION VARCHAR2(60) not null
VAR_CNT
VAR_LEN_CNT_SID <pk> NUMBER(2) not null
NAME VARCHAR2(30) not null
DESCRIPTION VARCHAR2(30) not null
DATA_ENCODE : 1
DATA_ENCODE_SID <pk> NUMBER(2) not null
NAME VARCHAR2(30) not null
DESCRIPTION VARCHAR2(30) not null
DATA_FMT
DATA_FMT_SID <pk> NUMBER(3) not null
NAME VARCHAR2(30) not null
DESCRIPTION VARCHAR2(30) not null
INTERNAL_MSG
INT_MSG_SID <pk> NUMBER(6) not null
NAME VARCHAR2(20) not null
DESCRIPTION VARCHAR2(30) not null
FIELD_SWITCH
INT_MSG_SID <pk,fk> NUMBER(4) not null
ORG_MSG_VER <pk,fk> NUMBER(2) not null
ORG_FIELD_NUM <pk,fk> NUMBER(3) not null
ORG_FIELD_DATA <pk> CHAR(32) not null
ORG_FIELD_LEN NUMBER(3) not null
MSG_VER <fk> NUMBER(2) not null
FIELD_NUM <fk> NUMBER(3) not null
FIELD_MAP
INT_MSG_SID <pk,fk> NUMBER(4) not null
ORG_MSG_VER <pk,fk> NUMBER(2) not null
ORG_FIELD_NUM <pk,fk> NUMBER(3) not null
ORG_FIELD_DATA <pk> CHAR(32) not null
ORG_FIELD_LEN NUMBER(3) not null
FIELD_LEN NUMBER(3) not null
FIELD_LEN_MIN NUMBER(3) not null
VAR_LEN_SID <fk> NUMBER(2) not null
VAR_LEN_CNT_SID <fk> NUMBER(2) not null
DATA_LEN_ENCODE_SID NUMBER(2) not null
DATA_ENCODE_SID <fk> NUMBER(2) not null
DATA_FMT_SID <fk> NUMBER(3) not null
FIELD_MAP_DATA CHAR(32) null
FIELD_DLMT CHAR(1) null

Fig 22

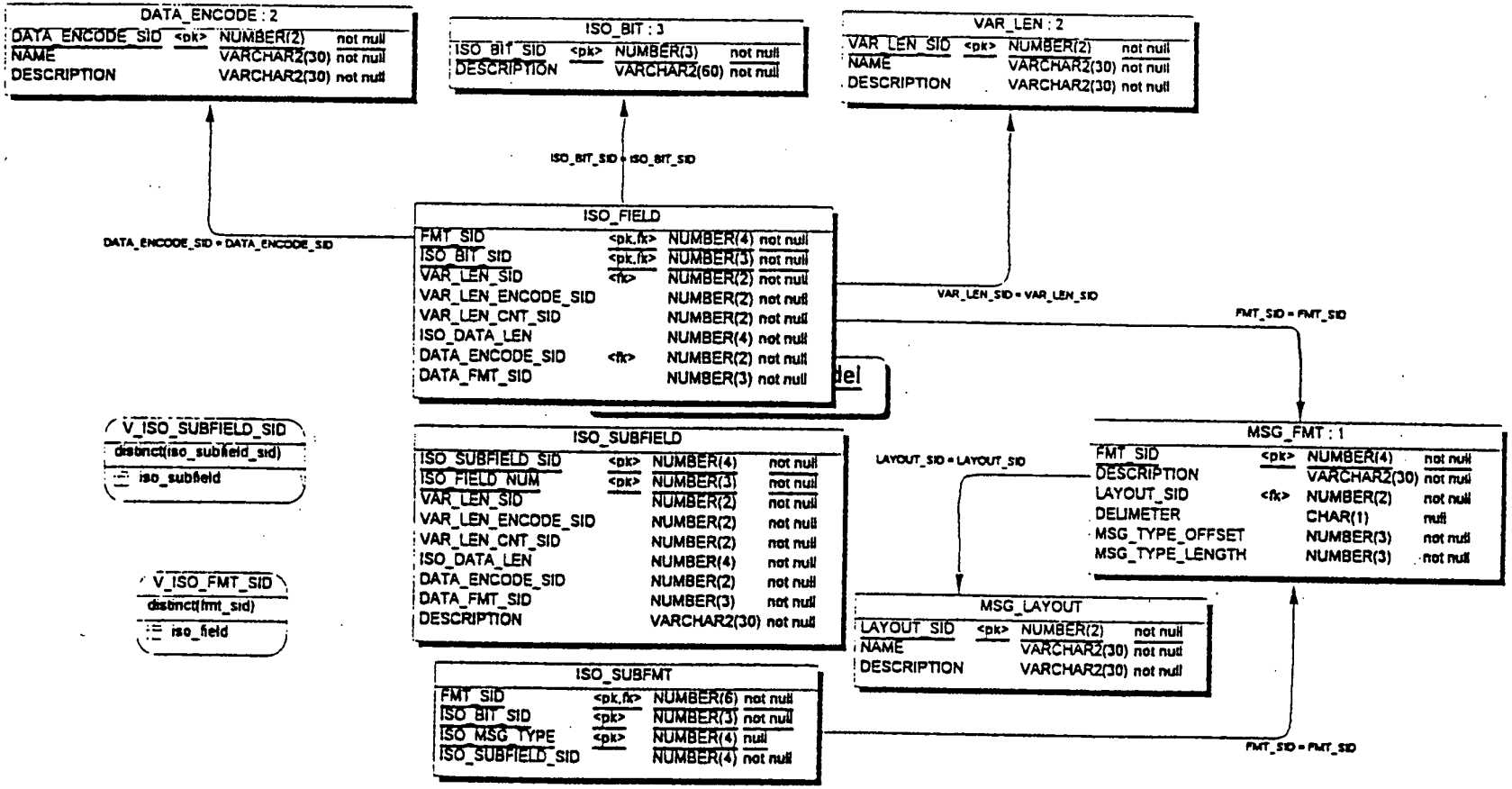

DATA_ENCODE : 2
DATA_ENCODE_SID <pk> NUMBER(2) not null
NAME VARCHAR2(30) not null
DESCRIPTION VARCHAR2(30) not null
ISO_BIT : 3
ISO_BIT_SID <pk> NUMBER(3) not null
DESCRIPTION VARCHAR2(60) not null
VAR_LEN : 2
VAR_LEN_SID <pk> NUMBER(2) not null
NAME VARCHAR2(30) not null
DESCRIPTION VARCHAR2(30) not null
ISO_BIT_SID = ISO_BIT_SID
DATA_ENCODE_SID = DATA_ENCODE_SID
ISO_FIELD
FMT_SID <pk,fk> NUMBER(4) not null
ISO_BIT_SID <pk,fk> NUMBER(3) not null
VAR_LEN_SID <fk> NUMBER(2) not null
VAR_LEN_ENCODE_SID NUMBER(2) not null
VAR_LEN_CNT_SID NUMBER(2) not null
ISO_DATA_LEN NUMBER(4) not null
DATA_ENCODE_SID <fk> NUMBER(2) not null
DATA_FMT_SID NUMBER(3) not null
VAR_LEN_SID = VAR_LEN_SID
FMT_SID = FMT_SID
V_ISO_SUBFIELD_SID
distinct(iso_subfield_sid)
iso_subfield
ISO_SUBFIELD
ISO_SUBFIELD_SID <pk> NUMBER(4) not null
ISO_FIELD_NUM <pk> NUMBER(3) not null
VAR_LEN_SID NUMBER(2) not null
VAR_LEN_ENCODE_SID NUMBER(2) not null
VAR_LEN_CNT_SID NUMBER(2) not null
ISO_DATA_LEN NUMBER(4) not null
DATA_ENCODE_SID NUMBER(2) not null
DATA_FMT_SID NUMBER(3) not null
DESCRIPTION VARCHAR2(30) not null
LAYOUT_SID = LAYOUT_SID
MSG_FMT : 1
FMT_SID <pk> NUMBER(4) not null
DESCRIPTION VARCHAR2(30) not null
LAYOUT_SID <fk> NUMBER(2) not null
DELIMETER CHAR(1) null
MSG_TYPE_OFFSET NUMBER(3) not null
MSG_TYPE_LENGTH NUMBER(3) not null
V_ISO_FMT_SID
distinct(fmt_sid)
iso_field
MSG_LAYOUT
LAYOUT_SID <pk> NUMBER(2) not null
NAME VARCHAR2(30) not null
DESCRIPTION VARCHAR2(30) not null
ISO_SUBFMT
FMT_SID <pk,fk> NUMBER(6) not null
ISO_BIT_SID <pk> NUMBER(3) not null
ISO_MSG_TYPE <pk> NUMBER(4) null
ISO_SUBFIELD_SID NUMBER(4) not null
FMT_SID = FMT_SID

Fig 23

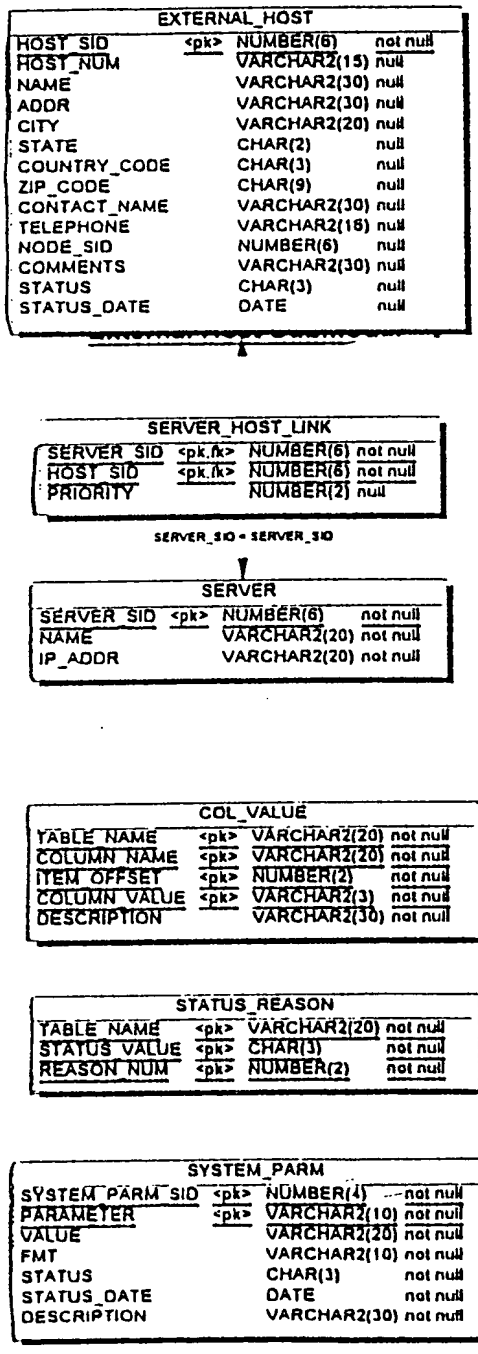
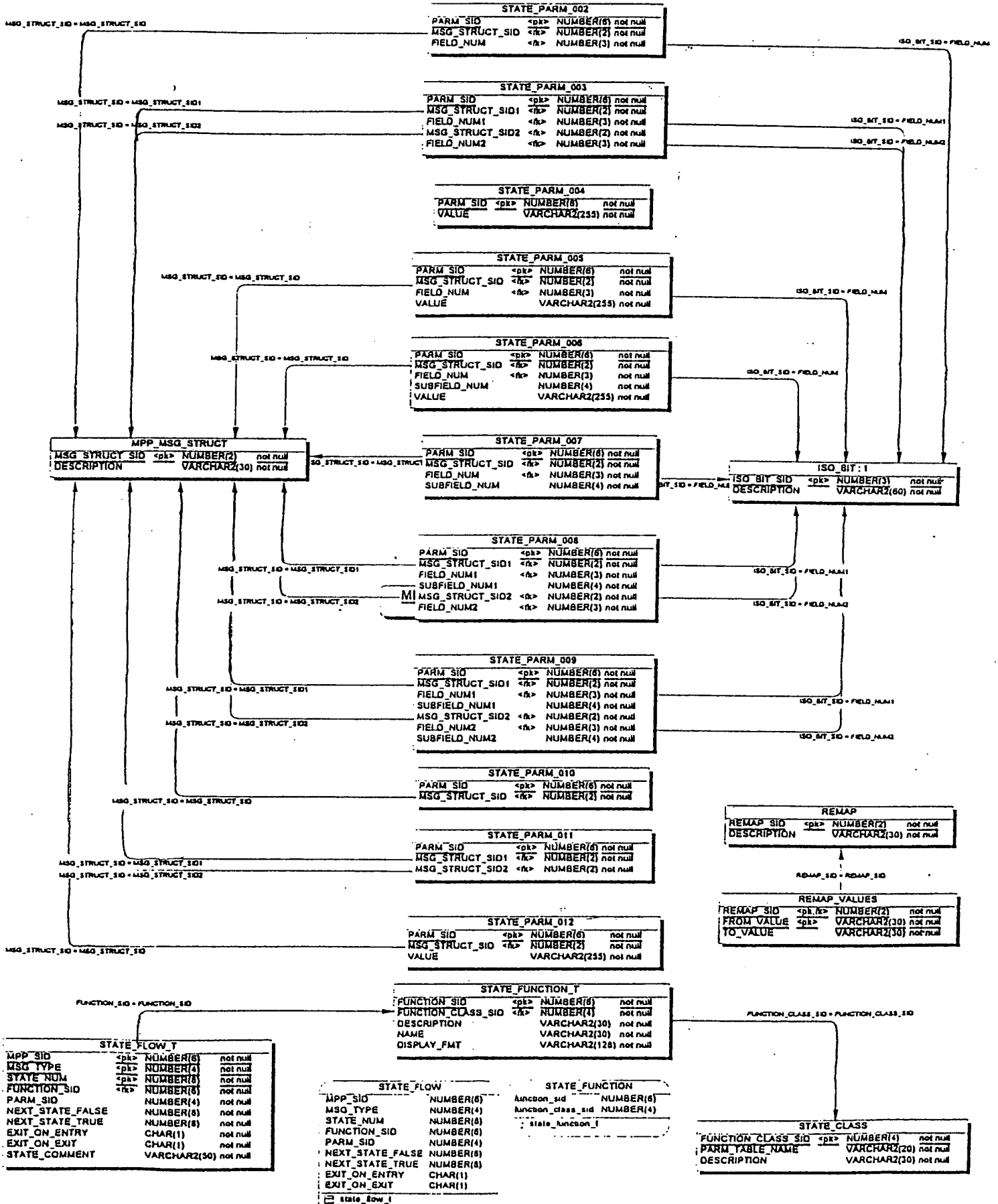

EXTERNAL_HOST
HOST_SID <pk> NUMBER(6) not null
HOST_NUM VARCHAR2(15) null
NAME VARCHAR2(30) null
ADDR VARCHAR2(30) null
CITY VARCHAR2(20) null
STATE CHAR(2) null
COUNTRY_CODE CHAR(3) null
ZIP_CODE CHAR(9) null
CONTACT_NAME VARCHAR2(30) null
TELEPHONE VARCHAR2(16) null
NODE_SID NUMBER(6) null
COMMENTS VARCHAR2(30) null
STATUS CHAR(3) null
STATUS_DATE DATE null
SERVER_HOST_LINK
SERVER_SID <pk,fk> NUMBER(6) not null
HOST_SID <pk,fk> NUMBER(6) not null
PRIORITY NUMBER(2) null
SERVER_SID = SERVER_SID
SERVER
SERVER_SID <pk> NUMBER(6) not null
NAME VARCHAR2(20) not null
IP_ADDR VARCHAR2(20) not null
COL_VALUE
TABLE_NAME <pk> VARCHAR2(20) not null
COLUMN_NAME <pk> VARCHAR2(20) not null
ITEM_OFFSET <pk> NUMBER(2) not null
COLUMN_VALUE <pk> VARCHAR2(3) not null
DESCRIPTION VARCHAR2(30) not null
STATUS_REASON
TABLE_NAME <pk> VARCHAR2(20) not null
STATUS_VALUE <pk> CHAR(3) not null
REASON_NUM <pk> NUMBER(2) not null
SYSTEM_PARM
SYSTEM_PARM_SID <pk> NUMBER(4) not null
PARAMETER <pk> VARCHAR2(10) not null
VALUE VARCHAR2(20) not null
FMT VARCHAR2(10) not null
STATUS CHAR(3) not null
STATUS_DATE DATE not null
DESCRIPTION VARCHAR2(30) not null

FIG 24

Fig 25


STATE_PARM_002
PARM_SID <pk> NUMBER(6) not null
MSG_STRUCT_SID <fk> NUMBER(2) not null
FIELD_NUM <fk> NUMBER(3) not null
STATE_PARM_003
PARM_SID <pk> NUMBER(6) not null
MSG_STRUCT_SID1 <fk> NUMBER(2) not null
FIELD_NUM1 <fk> NUMBER(3) not null
MSG_STRUCT_SID2 <fk> NUMBER(2) not null
FIELD_NUM2 <fk> NUMBER(3) not null
STATE_PARM_004
PARM_SID <pk> NUMBER(6) not null
VALUE VARCHAR2(255) not null
STATE_PARM_005
PARM_SID <pk> NUMBER(6) not null
MSG_STRUCT_SID <fk> NUMBER(2) not null
FIELD_NUM <fk> NUMBER(3) not null
VALUE VARCHAR2(255) not null
STATE_PARM_006
PARM_SID <pk> NUMBER(6) not null
MSG_STRUCT_SID <fk> NUMBER(2) not null
FIELD_NUM <fk> NUMBER(3) not null
SUBFIELD_NUM NUMBER(4) not null
VALUE VARCHAR2(255) not null
MPP_MSG_STRUCT
MSG_STRUCT_SID <pk> NUMBER(2) not null
DESCRIPTION VARCHAR2(30) not null
STATE_PARM_007
PARM_SID <pk> NUMBER(6) not null
MSG_STRUCT_SID <fk> NUMBER(2) not null
FIELD_NUM <fk> NUMBER(3) not null
SUBFIELD_NUM NUMBER(4) not null
ISO_BIT : 1
ISO_BIT_SID <pk> NUMBER(3) not null
DESCRIPTION VARCHAR2(60) not null
STATE_PARM_008
PARM_SID <pk> NUMBER(6) not null
MSG_STRUCT_SID1 <fk> NUMBER(2) not null
FIELD_NUM1 <fk> NUMBER(3) not null
SUBFIELD_NUM1 NUMBER(4) not null
MSG_STRUCT_SID2 <fk> NUMBER(2) not null
FIELD_NUM2 <fk> NUMBER(3) not null
STATE_PARM_009
PARM_SID <pk> NUMBER(6) not null
MSG_STRUCT_SID1 <fk> NUMBER(2) not null
FIELD_NUM1 <fk> NUMBER(3) not null
SUBFIELD_NUM1 NUMBER(4) not null
MSG_STRUCT_SID2 <fk> NUMBER(2) not null
FIELD_NUM2 <fk> NUMBER(3) not null
SUBFIELD_NUM2 NUMBER(4) not null
STATE_PARM_010
PARM_SID <pk> NUMBER(6) not null
MSG_STRUCT_SID <fk> NUMBER(2) not null
REMAP
REMAP_SID <pk> NUMBER(2) not null
DESCRIPTION VARCHAR2(30) not null
REMAP_VALUES
REMAP_SID <pk,fk> NUMBER(2) not null
FROM_VALUE <pk> VARCHAR2(30) not null
TO_VALUE VARCHAR2(30) not null
STATE_PARM_011
PARM_SID <pk> NUMBER(6) not null
MSG_STRUCT_SID1 <fk> NUMBER(2) not null
MSG_STRUCT_SID2 <fk> NUMBER(2) not null
STATE_PARM_012
PARM_SID <pk> NUMBER(6) not null
MSG_STRUCT_SID <fk> NUMBER(2) not null
VALUE VARCHAR2(255) not null
STATE_FUNCTION_T
FUNCTION_SID <pk> NUMBER(6) not null
FUNCTION_CLASS_SID <fk> NUMBER(4) not null
DESCRIPTION VARCHAR2(30) not null
NAME VARCHAR2(30) not null
DISPLAY_FMT VARCHAR2(128) not null
STATE_FLOW_T
MPP_SID <pk> NUMBER(6) not null
MSG_TYPE <pk> NUMBER(4) not null
STATE_NUM <pk> NUMBER(8) not null
FUNCTION_SID <fk> NUMBER(6) not null
PARM_SID NUMBER(4) not null
NEXT_STATE_FALSE NUMBER(8) not null
NEXT_STATE_TRUE NUMBER(8) not null
EXIT_ON_ENTRY CHAR(1) not null
EXIT_ON_EXIT CHAR(1) not null
STATE_COMMENT VARCHAR2(50) not null
STATE_FLOW
MPP_SID NUMBER(6)
MSG_TYPE NUMBER(4)
STATE_NUM NUMBER(8)
FUNCTION_SID NUMBER(6)
PARM_SID NUMBER(4)
NEXT_STATE_FALSE NUMBER(8)
NEXT_STATE_TRUE NUMBER(8)
EXIT_ON_ENTRY CHAR(1)
EXIT_ON_EXIT CHAR(1)
state_flow_I
STATE_FUNCTION
function_sid NUMBER(6)
function_class_sid NUMBER(4)
state_function_I
STATE_CLASS
FUNCTION_CLASS_SID <pk> NUMBER(4) not null
PARM_TABLE_NAME VARCHAR2(20) not null
DESCRIPTION VARCHAR2(30) not null

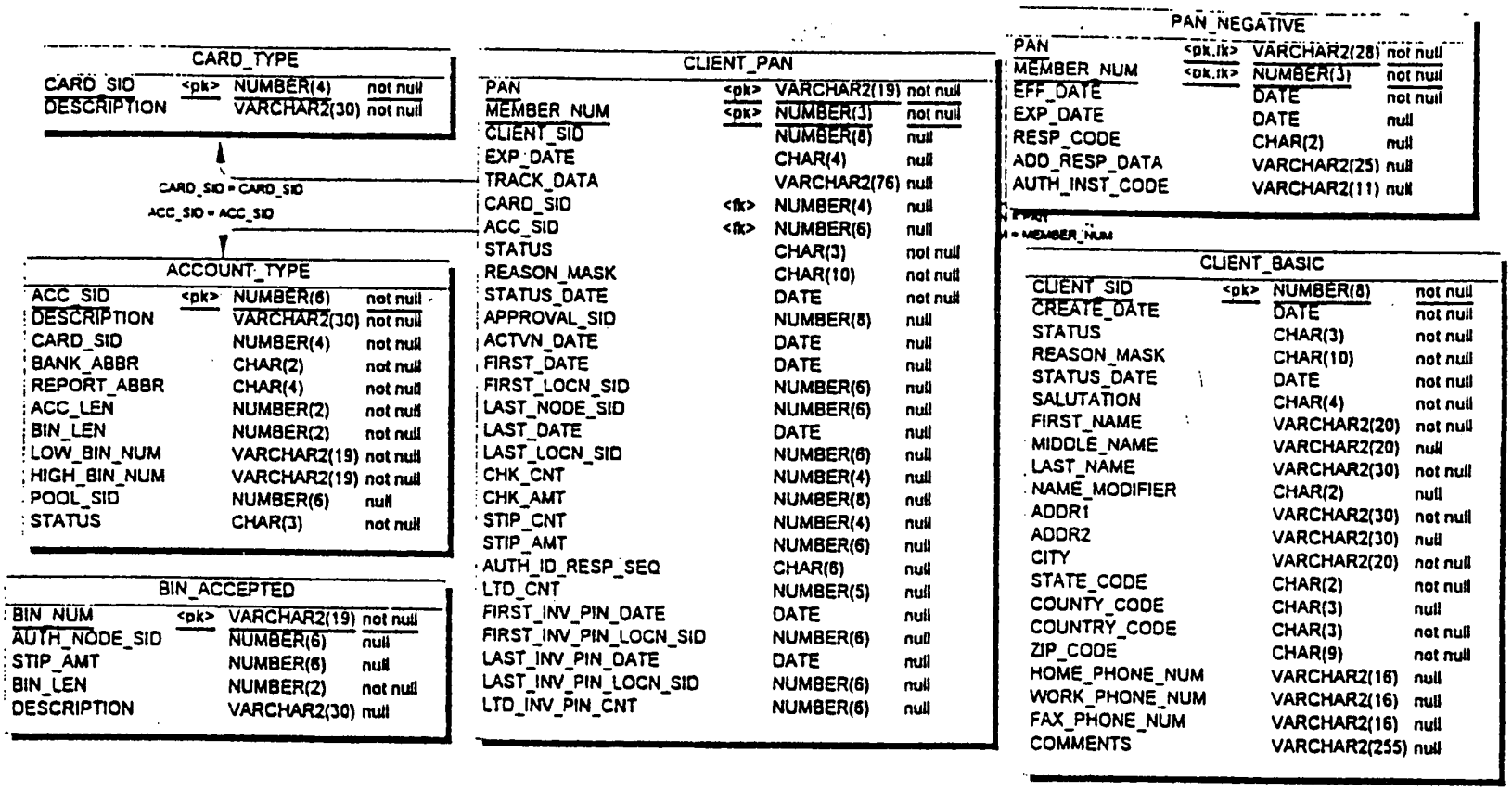

CARD_TYPE
CARD_SID <pk> NUMBER(4) not null
DESCRIPTION VARCHAR2(30) not null
CARD_SID = CARD_SID
ACC_SID = ACC_SID
ACCOUNT_TYPE
ACC_SID <pk> NUMBER(6) not null
DESCRIPTION VARCHAR2(30) not null
CARD_SID NUMBER(4) not null
BANK_ABBR CHAR(2) not null
REPORT_ABBR CHAR(4) not null
ACC_LEN NUMBER(2) not null
BIN_LEN NUMBER(2) not null
LOW_BIN_NUM VARCHAR2(19) not null
HIGH_BIN_NUM VARCHAR2(19) not null
POOL_SID NUMBER(6) null
STATUS CHAR(3) not null
BIN_ACCEPTED
BIN_NUM <pk> VARCHAR2(19) not null
AUTH_NODE_SID NUMBER(6) null
STIP_AMT NUMBER(6) null
BIN_LEN NUMBER(2) not null
DESCRIPTION VARCHAR2(30) null
CLIENT_PAN
PAN <pk> VARCHAR2(19) not null
MEMBER_NUM <pk> NUMBER(3) not null
CLIENT_SID NUMBER(8) null
EXP_DATE CHAR(4) null
TRACK_DATA VARCHAR2(76) null
CARD_SID <fk> NUMBER(4) null
ACC_SID <fk> NUMBER(6) null
STATUS CHAR(3) not null
REASON_MASK CHAR(10) not null
STATUS_DATE DATE not null
APPROVAL_SID NUMBER(8) null
ACTVN_DATE DATE null
FIRST_DATE DATE null
FIRST_LOCN_SID NUMBER(6) null
LAST_NODE_SID NUMBER(6) null
LAST_DATE DATE null
LAST_LOCN_SID NUMBER(6) null
CHK_CNT NUMBER(4) null
CHK_AMT NUMBER(8) null
STIP_CNT NUMBER(4) null
STIP_AMT NUMBER(6) null
AUTH_ID_RESP_SEQ CHAR(6) null
LTD_CNT NUMBER(5) null
FIRST_INV_PIN_DATE DATE null
FIRST_INV_PIN_LOCN_SID NUMBER(6) null
LAST_INV_PIN_DATE DATE null
LAST_INV_PIN_LOCN_SID NUMBER(6) null
LTD_INV_PIN_CNT NUMBER(6) null
PAN_NEGATIVE
PAN <pk,fk> VARCHAR2(28) not null
MEMBER_NUM <pk,fk> NUMBER(3) not null
EFF_DATE DATE not null
EXP_DATE DATE null
RESP_CODE CHAR(2) null
ADD_RESP_DATA VARCHAR2(25) null
AUTH_INST_CODE VARCHAR2(11) null
PAN
M = MEMBER_NUM
CLIENT_BASIC
CLIENT_SID <pk> NUMBER(8) not null
CREATE_DATE DATE not null
STATUS CHAR(3) not null
REASON_MASK CHAR(10) not null
STATUS_DATE DATE not null
SALUTATION CHAR(4) not null
FIRST_NAME VARCHAR2(20) not null
MIDDLE_NAME VARCHAR2(20) null
LAST_NAME VARCHAR2(30) not null
NAME_MODIFIER CHAR(2) null
ADDR1 VARCHAR2(30) not null
ADDR2 VARCHAR2(30) null
CITY VARCHAR2(20) not null
STATE_CODE CHAR(2) not null
COUNTY_CODE CHAR(3) null
COUNTRY_CODE CHAR(3) not null
ZIP_CODE CHAR(9) not null
HOME_PHONE_NUM VARCHAR2(16) null
WORK_PHONE_NUM VARCHAR2(16) null
FAX_PHONE_NUM VARCHAR2(16) null
COMMENTS VARCHAR2(255) null

Fig 26

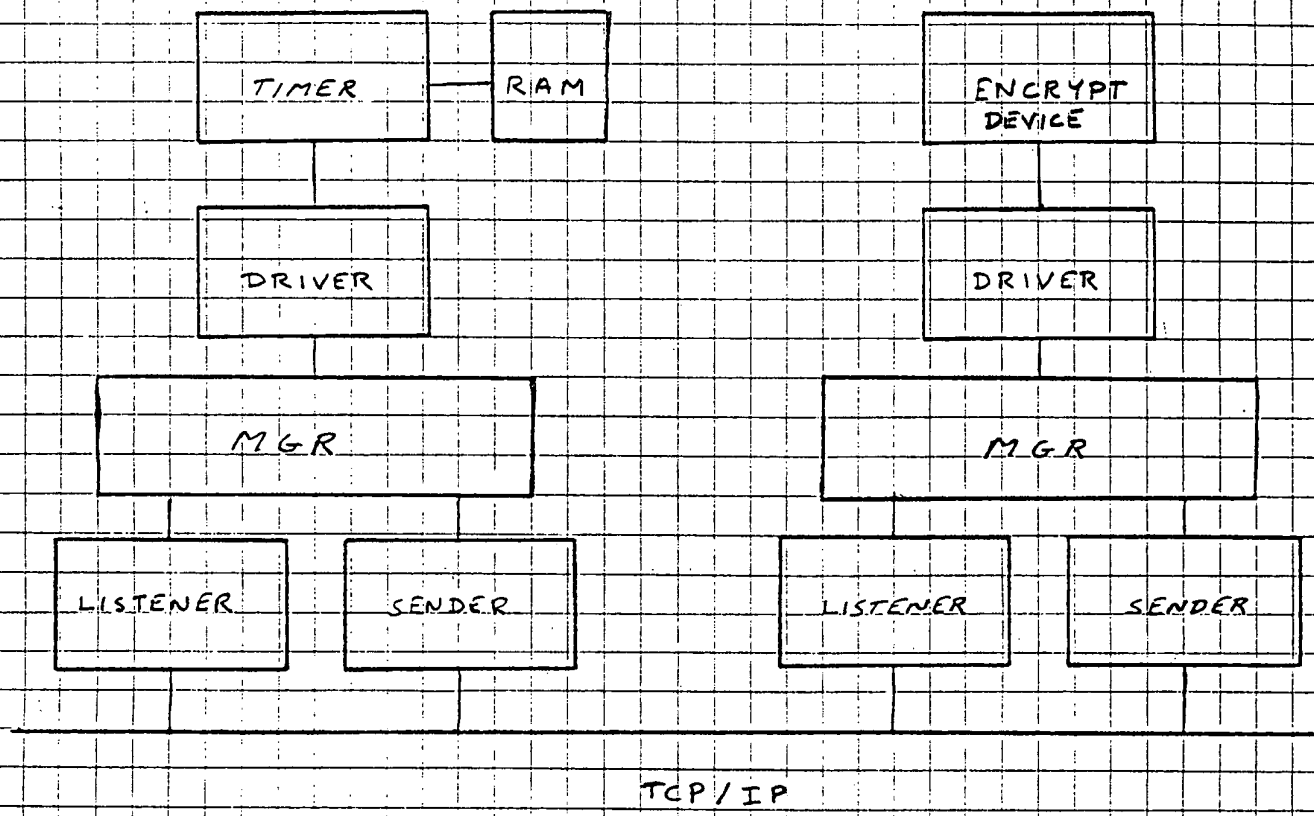

TIMER
RAM
ENCRYPT DEVICE
DRIVER
DRIVER
MGR
MGR
LISTENER
SENDER
LISTENER
SENDER
TCP/IP

Fig 27

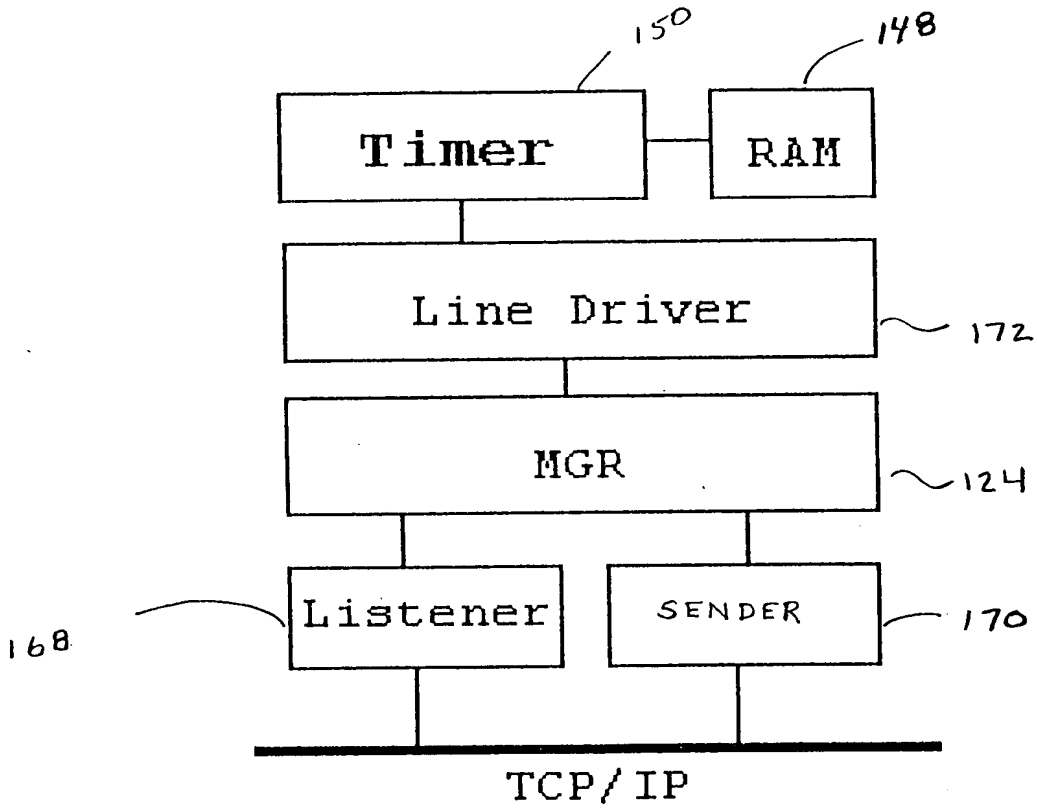
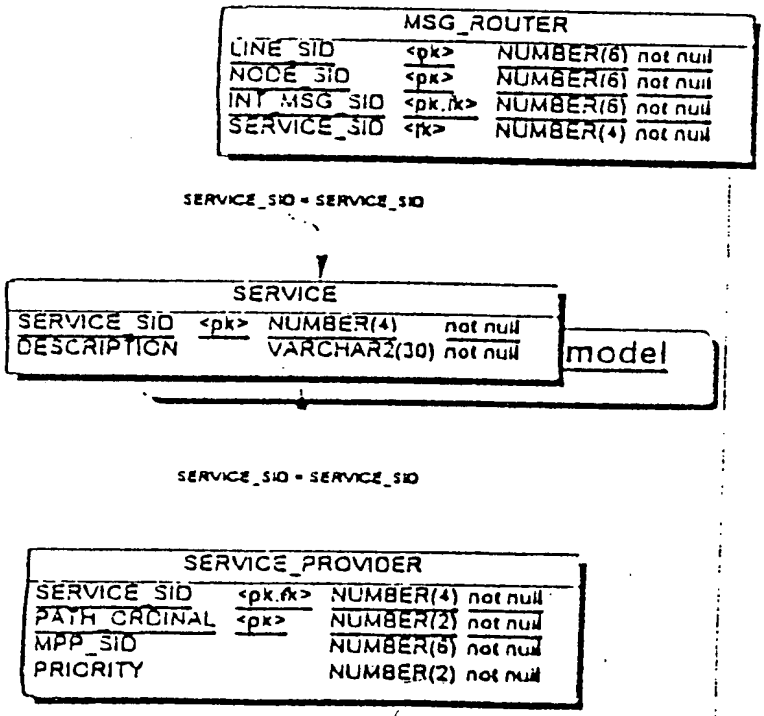

150
148
Timer
RAM
Line Driver
172
MGR
124
168
Listener
SENDER
170
TCP/IP

Fig 28

FIG 29


MSG_ROUTER
LINE_SID <pk> NUMBER(6) not null
NODE_SID <pk> NUMBER(6) not null
INT_MSG_SID <pk,fk> NUMBER(6) not null
SERVICE_SID <fk> NUMBER(4) not null
SERVICE_SID = SERVICE_SID
SERVICE
SERVICE_SID <pk> NUMBER(4) not null
DESCRIPTION VARCHAR2(30) not null
model
SERVICE_SID = SERVICE_SID
SERVICE_PROVIDER
SERVICE_SID <pk,fk> NUMBER(4) not null
PATH_ORDINAL <pk> NUMBER(2) not null
MPP_SID NUMBER(6) not null
PRIORITY NUMBER(2) not null

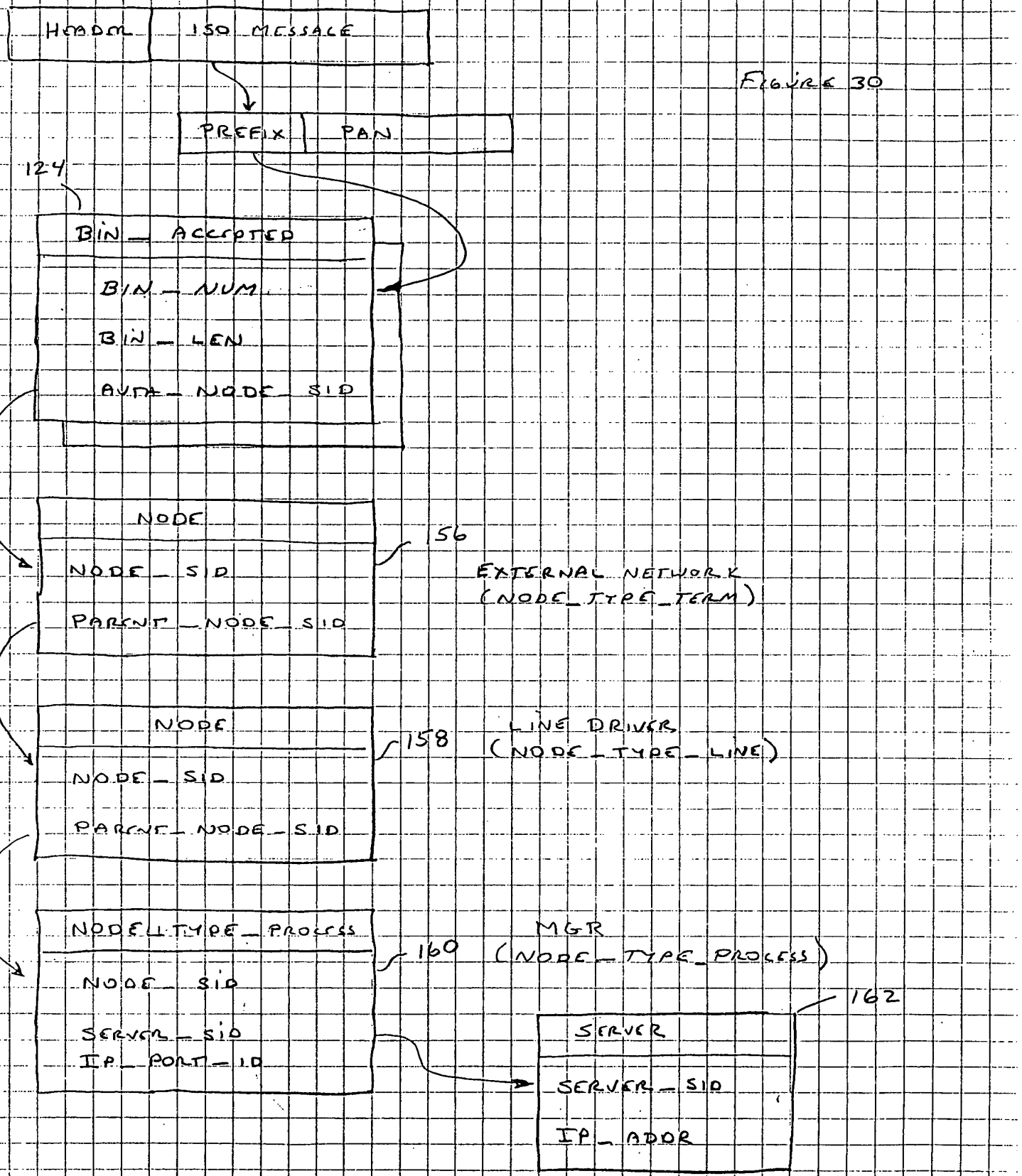

HEADER
ISO MESSAGE
PREFIX
PAN
124
BIN_ACCEPTED
BIN_NUM
BIN_LEN
AUTH_NODE_SID
NODE
NODE_SID
PARENT_NODE_SID
56
EXTERNAL NETWORK
(NODE_TYPE_TERM)
NODE
NODE_SID
PARENT_NODE_SID
158
LINE DRIVER
(NODE_TYPE_LINE)
NODE_TYPE_PROCESS
NODE_SID
SERVER_SID
IP_PORT_ID
160
MGR
(NODE_TYPE_PROCESS)
SERVER
SERVER_SID
IP_ADDR
162

FIGURE 30